JUNE 1981

SOFTALK

Chairman John Haller
Editor Margot Comstock Tommervik
Managing Editor Al Tommervik
Associate Editor Craig Stinson
Art Director Kurt A. Wahlner
Contributing Editors Roger Wagner
Jim Merritt
Peter Olivieri
Frank Malone
William Depew
Assistant Editor Robert Koehler
Operations Mary Sue Rennells
Production Bill Barrett
Administrative Assistant Kimberly Curling
Circulation David Hunter
Ron Rennells
Systems William V. R. Smith
Advertising Sales Al Tommervik
(213) 980-5074 Margot Tommervik

Cover photo of Apple president Mike Markkula by Kurt Wahlner.

Composition by Photographics, Hollywood, California. Printing by California Offset Printers, Glendale, California.

Apple is a registered trademark of Apple Computer Inc., Cupertino, California. UCSD Pascal is a trademark of the University of California at San Diego. *VisiCalc* is a trademark of Personal Software, Sunnyvale, California.

Subscriptions: Complimentary to all owners of Apple computers in the USA. Multiple or non-Apple-owner subscriptions for one year: one, $10; two through five, $8 each; more than five, $6 each.

Back Issues: $2 through February 1981; $2.50 thereafter. November and December 1980 issues are sold out. January and March 1981 are in short supply.

Moving? Send new address and old to Softalk Circulation, 11021 Magnolia Boulevard, North Hollywood, CA 91601; telephone, (213) 980-5099.

Softalk, Volume 1, Number 10. Copyright © 1981 by Softalk Publishing Inc. All rights reserved. ISSN:0274-9629. *Softalk* is published monthly by Softalk Publishing Inc., 11021 Magnolia Boulevard, North Hollywood, CA 91601, telephone, (213) 980-5074. Controlled circulation postage paid at Los Angeles, CA. Postmaster: Send address changes to Softalk, 11021 Magnolia Boulevard, North Hollywood, CA 91601.

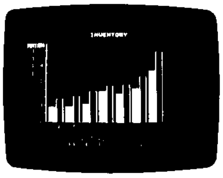
ADVERTISERS INDEX

Advanced Business Technology 44
Artsci 78
Aurora 25
Avant-Garde 27
Axiom 46
Beagle Bros.............11
Bite-Soft 52
The Book 1981 54
Broderbund Software 32
BudgeCo. 82
California Pacific ....Cover 2
Compdent 8
Computer Station 73, 75
Continental Software 36
Co-op Software 14
Creative Computing 10
Dakin5.............15
Data Transforms 26
Edu-Ware 41
FSI 40
GS Computer Enterprises 20
Highlands Computer Services............. 9
High Technology.............83
Howard Software 29
Innovative Design Software.............55
Intelligent Computer Systems 16
Interactive Microware 24, 50
Math City 79
MicroCom 67
Micro Lab 17, 19, 21
Micromate.............66
Microsoft 30
Micro-Ware Distributing.............74
Moneydisk.............48
MUSE Software 12, 33, 60, 64
Mytopia Software Institute 7
On-Line Systems.............61, 81, Cover 4
Rainbow Computing 18
RTR Market Charter 56
Sirius Software 45, 47, 49, 51, 53
Softape 68
Softsel 6
Software Publishing Corp. 3
Software Technology for Computers 57
Sophisticated Microsystems 31
Southwestern Data Systems 69
Spectrum Software 13
Stellation Two 39
Stoneware 28, 65, 77
Strategic Simulations ....Cover 3
Street Electronics 72
Synergistic Software............. 37, 70
United Software of America 42-43

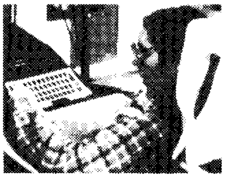
CONTENTS

Exec Apple: New President Mike Markkula
AL TOMMERVIK 4

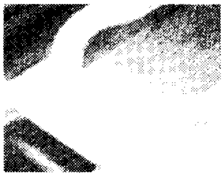
Fair Days 22



At 3.3, VisiCalc Spawns a Family 31



An Apple for Leslie
MELISSA MILICH 34



Going Forth
CRAIG STINSON 38

FEATURES

SPECIAL: Expanded Summer Computing Marketalk Reviews 48
DEBUT: Mind Your Business, Introduction .... PETER OLIVIERI 59
The Miraculous Modem, Part 3 .... JEFF MAZUR 66
DEBUT: Beginners' Corner .... CRAIG STINSON 71
The Double Life of Lord British 80

DEPARTMENTS

More Unknowns: Contest 2
Straightalk 7
Open Discussion 8
Basic Solution: William V. R. Smith 14
Assembly Lines: Roger Wagner 16
Ventures with VisiCalc: Otto K. Wetzel, Jr. . 26
Marketalk: News 44
Tradetalk 60
The Pascal Path: Jim Merritt 74
Softalk Presents the Bestsellers 83

PREVIEWS

Sizzling July . . . Exec Strategic Simulations . . . T-Shirt Country, Computer-Style . . . Are you ready for a hard disk drive? . . . And more . . .

CONTEST: THE LOGICAL SOURCE

The Unknown A, February's contest, was the most popular *Softalk* contest yet; it was also too easy—almost everyone caught on to Apple's address. So, this month, the contest is another unknown A, with a little more to it. It's not really a whole lot harder, but it gives you a lot more opportunity to get mixed up. And it brings a new element, a bit of word play, for those who prefer English to math.

There are two levels of winning the contest. Both levels must name the Logical Source of the value A. This is the name—a proper noun—of a person, place, or thing. The grand prize winner must also give the value of the letter A.

The nine letters that make up the name of the Logical Source have been given values according to their positions in the alphabet. Those values were added together to get the value, A. You must work this path backward.

How To Play

0. If you wish to ignore all the formulas and rigmarole, you can try to guess the correct identity of the Logical Source from the Direct Clues in part one and send it in. You'll be eligible for a second-level prize. For a first-prize chance, you'll have to do the rest, however.

1. Determine the values for the letters in the formula (part two) by figuring out the clues in part three.

2. Work the formula in part two to get A.

3. Use A in the miniformulas of part four to determine the alphabet values of the letters that were added to make A.

4. Thinking of the alphabet as a circle, assume each letter to have the value of its position in the circle, starting with A as 1; this would be very straightforward except that, during the night while the alphabet was waiting to be used to build this puzzle, someone who didn't want you to win put a hex on the alphabet and changed all its values. Fortunately, the change was consistent, and, by reading carefully for clues, you will be able to figure out the new values. When you figure out what letters the values you got in part four represent under the hexed alphabet, you'll have all the letters of the name of the Logical Source.

5. Rearrange (anagram) these letters into the name that fits the Direct Clues to the name of the Logical Source.

6. Send the name of the Logical Source and, for the grand prize, the value of A, to Softalk Source, 11021 Magnolia Boulevard, North Hollywood, CA 91601.

First prize is $100 in goods from *Softalk* advertisers. Second-level prize is $50 worth of *Softalk* advertisers' products. In case of ties, contestants will be at the mercy of Apple's random generator.

Here's the puzzle.

Part 1: Direct Clues

1. Without the contributions of the Logical Source, computers might not yet exist.
2. Shipping a fortune.
3. The Logical Source's major predecessor was big on forms.
4. Greece.
5. Another formula using A.
6. Cause of a temperamental opera star's being at sea.
7. Philosophizing.

Part 2: The Formula for A

$$A = \frac{(Qz^{P} + \sqrt{x})^{P}}{u} \cdot y + \frac{v^{P} - n^{P}}{w} + \sqrt[n]{v} \cdot m$$

Part 3: The Formula Values

Q = Brahms symphonies or Rachmaninoff piano concertos. Lucky leaves. Control-D.
z = Marx Brothers. Orange. Little Women + Nancy Drew.
p = A brace or yoke. If by sea. Bits in a quarter.
x = Crayons in the big box. Gadolinium. Combined age of quadruplets on their sweet birthday.
u = Half your phalanges. Base of any number system as represented in that system.
y = Year Vikings and French gave England the ol' one-two. $42B.
v = Poe's gables plus Clark. Atlas's daughters. Top row, middle key.
n = Volumes in *The Foundation, U.S.A.,* or *The Octopus.* Number involved in unlucky cigarette lighting habit.
w = Polk. Jack. 0000 1011.
m = Great wealth on the Bowery. Winning margin in time when place horse follows winner by two lengths. Bargain basement discount.

Part 4: Alphabet Letter Values

#1 — = INT (A / 5)—4
#2 — = (A + 3) / (10 + 1)
#3 — = INT (A / [=1]) * 4 / 3
#4 — = [#1] + [#3]
#5 — = (A * 2 + [#1] + [#4])$^{0.125}$
#6 — = [#2] / [#5]
#7 — = SQR (A—[#1]—9)
#8 — = (A - [#5]) / [#6]
#9 — = [#4] / [#6] * [#5]

If #5 looks hard to you—ask your Apple!

Part 5: Anagram

Fill in the numbered blanks with the letters derived from the alphabet values determined in part four.

— — — — — — — — —
#1 #2 #3 #4 #5 #6 #7 #8 #9

Rearrange the letters to form your answer for the Logical Source.

Fill in the entry form or facsimile.

Mail to Softalk Source, 11021 Magnolia Boulevard, North Hollywood, CA 91601, by July 15, 1981.

A = ___

The Logical Source is — — — — — — — — —

Name:______

Address:______

City/State/Zip:______

The prize I'd like if I win is______

My local dealer:______

My autograph:______

Exec Apple: New President Mike Markkula

BY ALLAN TOMMERVIK

Mike Markkula is a man who believes in plans. This belief is based on empirical evidence—his plan to provide for his retirement by the age of thirty-five was realized by the time he was thirty and his plan for Apple Computer Inc. is testimony to the efficacy of business planning.

The reputation of Apple's business plan has reached legendary proportions within the company. Relative newcomers to the company point with awe to the fact that before the first one thousand computers had been sold, company executives had provided a plan that, foresaw, almost to the exact employee and the exact square foot, what their requirements would be three years later.

Even in the volatile arena of commerce it figures that such prescience would garner its own reward; and so it seems to have transpired for the executive team and employees at Apple.

The company's public stock issue late last year appropriately rewarded all Apple employees who had worked to implement the business plan and who had the foresight to get involved in the company's stock option plan.

Any number of individuals—from Steve Wozniak and Steve Jobs, the inventors of the original Apple, to the entire dedicated staff of the company—deserve credit for this success. But as important as any element in the Apple success story is that of Markkula, his foresight, and his plan.

The Markkula Retirement Plan. In introducing Markkula to an assemblage of computer retailers at Apple Expo in Los Angeles, Phil Roybal, manager of editorial services, indicated as much when he credited Markkula with being the first to see such potential in the Apple that he provided seed capital to make it go.

Roybal also credited Markkula's strength of belief in the personal computer concept and in Apple with being a prime force in the subsequent success of the company.

And then, of course, there was the plan.

Markkula has been a constructor of plans. A man of widely diverse interests from such physical activities as tennis and furniture making to such more sedentary pursuits as guitar playing and venture capital investment, he saw early that attention to such interests was not best served within the framework of the customary career in industry.

So he constructed a plan while still in college that would theoretically permit his retirement at the age of thirty-five. He followed the plan faithfully through his career as an engineer at Hughes and in marketing positions with Fairchild Semiconductor and Intel.

The gratifying result was that he had achieved all the goals of the plan by age thirty. But he was so enjoying his work at Intel at that time that he postponed his life of leisure for an additional two years.

When he did opt for surcease from the stress of industry, he constructed a list of things he would like to accomplish or pursue. It numbered more than fifty items. Two years later the list was longer yet, although the items then on the list were different from the original items.

During that time, in keeping with his musical interests, he invented a wheel that would convert any chord into the appropriate fret positions on the guitar and show all possible positions for that chord.

Helping Hand Mushroomed into Second Career. And he kept his hand in industry by helping fledgling businesses construct business plans. It was through this activity that his plans for lifelong retirement went awry.

An acquaintance introduced him to Jobs and Wozniak, identifying them as two youngsters with a hot product who were badly in need of a business plan.

Markkula agreed to showing them how to construct a viable business plan. But after meeting and talking to them, he realized that "Steve and Woz are not the kind of people who ever commit anything to paper." So he agreed to undertake the writing of the business plan for Apple.

It will come as no surprise to anyone who has ever committed any idea to paper that Markkula became enamored with the plan he had wrought. Of course, his prior success at structuring business plans gave him a more empirical base for this feeling than many authors have; nevertheless, when it became apparent that implementation of the business plan likely would be wanting because of a shortage of capital and marketing expertise, he provided both—supplying the first venture capital for the company and abandoning his retirement plan to take a hand at marketing the Apple.

Since those early days, Markkula's influence has been pervasive in the company, first as chairman of the board and now as president and chief executive officer. His move to the presidency, succeeding Mike Scott, caused eyebrows to be raised, coming as it did soon after the most un-Applelike action of laying off some forty employees and on the heels of a rather dismal unveiling of the Apple III.

Speculation has been rampant that Scott and "forty engineers" were being made the whipping boys for the lack of success of the III.

Markkula is a genuinely nice person, uncharacteristically humane for the chief honcho of a three hundred million dollar organization. Testimonials to his character, thoughtfulness, openness to new ideas, and willingness to delegate responsibility come unsolicited from every corner of Apple. So his pained expression at hearing such reports is real.

While admitting that the introduction of the Apple III has been a disappointment, Markkula denies any link between that disappointment and the layoffs and executive suite changes.

The Story Behind the Rumors. The layoffs came at the end of a year of improbable expansion by the company, with nearly one thousand new employees added. Clearly the laws of probability defy any company to add that many new employees without making some errors of judgment and placing some qualified people in areas outside their main expertise.

That, per Markkula, was what happened at Apple and precipitated the layoffs. The personnel changes affected all areas of the company, so the notion that forty engineers got the axe represented a total misreading of the occurrence. The event probably caused more consternation because Apple has earned the image of a company with high morale that is good to work for.

It was likewise puzzling to those who knew that Apple was then and is still hiring at all levels in all divisions of the company. Within a month of the layoffs, Apple had added more than forty new employees.

Grommet manufacturers and garment industry firms, organizations unknown for an esprit de corps such as that at Apple, regularly lay off hundreds without any note whatsoever being taken.

Likewise, the executive suite shift—executive suite representing real hyperbole for the modest offices occupied by Apple's top team—was apparently neither revolution nor evolution, but merely a reorganizing of the talents available at the top to address the problems now extant, with Jobs taking Markkula's spot as chairman of the board and Scott stepping up to vice-chairman.

As he has in the past, Markkula still takes every opportunity to explain that titles are not a primary consideration among the executive team at Apple.

"We gave serious consideration to doing away with all titles here, but that idea had two drawbacks. Primarily, we found that the outside world had a great deal of difficulty coping with the concept. Secondly, new employees to the company would have difficulty understanding where they fit in without some indication of hierarchy."

Apple's solution to the problem is to keep titles to a minimum and to make them as definitive as possible.

But, at the highest level, the emphasis is on problem solving and not on titles, with the result that executive changes there have less meaning than at other companies where palace coups or kicking an exec upstairs are regular occurrences.

Anticipating the Future—and Future Competition. In the present instance, it was felt that Scott's talents were more desperately needed in the area of advance planning and long-range projects. While long-range planning has an honored place at most companies of substance, it's even more vital at Apple, where the dynamics of technology and the marketplace make planning not merely an adjunct of success but the very essence of it.

The entrance of Adam Osborne and three Japanese firms into the personal and desktop computer competition and the possible entry of such giants as IBM and Digital Equipment Corporation into the personal computer field mandates careful charting of the future course.

Likewise, such innovations as Intel's 64K RAM chip and Hewlett-Packard's 450K microprocessor, developed but still untested, augur a changed technological environment to which every company in the industry must address itself.

Markkula believes that, unlike the radio, television, watch, and calculator industries, Apple and the other American microcomputer firms have a good chance of withstanding the onslaught of Japanese technology and product.

His analysis first recognizes the differences between those products that are now predominantly Japanese made and computers—the other products were essentially impulse purchases that had little or no need for follow-on support from the seller.

Computers, on the other hand, require considerable after-purchase service and support, and the Apple franchise network is viewed as a bulwark against significant inroads because of its strength in the areas of the maintenance, repair, and customer service.

Markkula believes that fully 50 percent of the revenues to be derived from the sale of a microcomputer stem from follow-on business. And the relationships and business practices already in place at the Apple franchises should stand Apple in good stead.

The new Apple president also believes that the large number of smaller companies who are supporting the Apple with peripherals and software places Apple in a strong competitive position. At Apple Expo, he cited the existence of the Apple as being responsible for the birth and growth of one hundred to one hundred fifty smaller companies. [Markkula may be using certain qualifiers to determine that number. *Softalk* has received expressions of interest in advertising from more than seven hundred fifty firms and individuals who are intent on the commercial marketing of product for the Apple.]

The Saga of Apple III. For all the success Markkula and Apple have had with planning, not everything goes according to plan. The introduction of the Apple III is a clear case in point.

Electronic glitches and marketing misjudgments have marked the advent of the III. The original design of the III included sockets in the motherboard that were not up to their assigned tasks. Expansion and contraction caused by the heating up and cooling down of the system would cause the chips to pop out of these sockets.

Apple rectified the problem by replacing the sockets with a new version that sits higher and by inserting the chips with greater force. This solution has caused its own, albeit minor, problem. Chips inserted with greater insertion pressure have a

GOTO 62

STRAIGHTALK

Summer is starting, students are commencing, and we at *Softalk,* as you read this, are beginning to put together the last issue of our first year. It's been said before—we couldn't have done it without you. Thanks for all your letters and calls, all your suggestions and your enthusiastic participation in our polls and contests, and for your loyal readership. Thanks, especially, for your willingness to try our advertisers' products, to support them so they could support us in our efforts to bring you free a fun magazine about your favorite computer.

But this is not an ending—we're just making way for the year two; so on all counts, as so many of you have written us, keep it coming! We chose to write about our anniversary early, from your point of view, just so you would have the opportunity to take part in planning our second year. Let us know what you'd like to see in *Softalk*'s second volume.

Some plans are already laid. We have begun six tutorial columns since September, two in this issue. And there are three more planned yet to begin: one on using the SoftCard, one on programming in Applesoft, and one on graphics. We'll continue our regular coverage of hardware and software. And we'll look with better than ever resources for stories and people: stories of unique uses of Apples by ordinary people, and colorful people who make ordinary Apple applications seem unique.

Each of the two columns beginning in this issue has been in the works several months. Mind Your Business will attempt to help you teach your Apple to do just that. Dr. Peter Olivieri teaches business and microcomputing with Apples at Boston College in Massachusetts; he acts also as consultant to businesses using computers; and he'll be writing regularly for *Softalk.*

Beginners' Corner deals with all the obvious things that everyone's supposed to assume and no one really understands until they've had their Apples six months or so, and it takes a shot at exorcising some of the lurking, if minor, fears about harming your computer. The first installment is designed to get the brand-new Apple owner running programs without having to read six manuals first. Whether in a business or at home, the first thing you want when you get your first computer is to see it run! Then, when you're bricked out, double Apple-visioned, and on downright friendly terms with your computer, is the time to figure out how to work it. If you're a veteran Apple owner—say, of four or five months' duration—perhaps you can use your Beginners' Corner to help a friend who's just bought an Apple or who's a little afraid about learning how to run one.

Also this month, check out our extra-long summertime computing review section.

Once again, thanks for enjoying *Softalk.*

MCT

OPEN DISCUSSION

Home Money Reminder

To clarify matters for your readers, *Home Money Minder* does not require a 132-column printer—an 80-column printer will do. Also, if the program is configured properly for the Apple parallel card, Apple's video should be properly disabled and screen wrapping will not occur.

One other point: The coauthor of *Home Money Minder*, Stephen Pollack, was not mentioned in the review.

Robert B. Schoenburg, Continental Software, Culver City, CA

Yachting Programs Are Yare

The United States Yacht Racing Union, the National Sports Authority for the racing sailor, is embarked on a program to develop a new Race Management Manual for use by race committees everywhere.

One section of the looseleaf formatted manual (or handbook) will be devoted to various computer and calculator programs and other such aids.

Already we have received a few programs for computers such as the one on the rules and several for scoring multiclass regattas, etc.

We earnestly solicit any and all programs readers might have developed relating to sailing, race scoring, handicapping, measurement rules, and the like.

A library of such contributions is being maintained at the union's headquarters and contributions should be sent there: USYRU, P.O. Box 209, Newport, RI 02840.

The listing of the programs in the library will be included in the manual and its frequent updates, with appropriate credit to the authors and contributors.

Any questions or comments should be sent to my attention.

P.S. I have written to Dr. Puckett for his programs ["A Seafaring Apple," April 1981].

Evans M. Harrell, USYRU Race Management Committee, Marietta, GA 30060

Registering a Problem

I have been trying to figure out how to use my Apple to process data from Sweda cash registers. If you have any info as to how I can do it, please send it to me.

Annette Herron, Troy, NY

After-Dark Hobbyists

We would like to hear from anyone who might be interested in subscribing to a network of hobby computers in which the central system would place calls to all the other users during the evening hours (reduced rates).

The system would be similar in nature to many of the bulletin board systems now available, but messages would be delivered and picked up by the central system instead of the users having to call in for them.

We don't know if such a system will turn out to be feasible but we would appreciate it if anyone with an interest in investigating the possibilities would send any comments, questions, or suggestions, along with a stamped, self-addressed envelope, to: Michael Witt, P.O. Box 55686, Valencia, CA 91355.

We can also be reached via modem at: (805) 255-6445.

Donna and Michael Witt, Valencia, CA

The Capable Cassette

Yes, I know: you're not a software publishing magazine. And yes, I know (it is claimed by some, anyway) that only 10 percent of Apple owners are diskless. But, honestly, a machine with disk really is no more capable than one without! Anyway, the program listing might be of interest. Clearly, being so Apple specific, such material is not of interest to the general computer magazines.

The program may not appear to do very much. Actually, though, it is the core capability for a whole raft of interactive application programs. It's good for any situation where the user wishes to change the contents of data without a new RUN command (remember, RUN reinitializes all variables; you cannot interrupt a program, change data, and then type CONTinue, either).

A few examples of applications: Mailing lists. Inventory control. Checkbook and financial data base programs. Multiple parameter engineering applications. The same approach will work with a disk, of course, but is less relevant. Incidentally, I notice that Radio Shack does have available a cassette-loaded mailing list program, whereas Apple does not. It is time to challenge the disk-o-chauvinists in Appleland!

Although the advantages of a disk cannot be disputed, many users get along without one for some time before making the additional investment. Program storage and input/output using tape have acquired a bad reputation for reliability and speed, but, with the use of inexpensive data quality C-10 cassettes and with the higher transfer speeds of systems such as the Apple, tape is an acceptable medium for many applications.

Nevertheless, the market does not yet offer application programs that can handle large amounts of data without being disk-based. This is because it is not easy to set up a file directly in Basic.

This Applesoft program establishes a 20,000-character file capability, organized as 500 blocks (lines) of 40 characters each. The utility subroutine (at line 800 for input, 900 for output) are the keys to using such a file. Because these subroutines involve direct (PEEK and POKE) memory accesses to specific memory locations, it is improbable that this program would work on any home computer other than the Apple II Plus. You need 32K of memory for this size file.

The reason for having 40 characters per line will be obvious when you key in the program. By using the repeat key, space bar, and cursor, the entry of 500 data lines should not be too tedious. Simply type in successively numbered DATA statements with exactly 40 spaces after the word DATA. Thus, when you are ready to RETURN statement number 11, the cursor is located thus:

```
10 DATA

11 DATA
       ▮
```

To this basic utility package you may now append whatever specialized programs you desire. For example, I am developing a mailing list package that uses this utility to handle up to 166 addresses (name, street address, city/state/zip, using three file lines per address). The principle that *must* be followed in any such application program is that *all* file accesses (input and output) must occur *only* via the subroutines at lines 800 and 900. *Never* enter the data file directly via another subroutine! When inputting or outputting, the variable FM identifies which file number (1 to 500) you desire. The string variable IN$ transfers data to the file when subroutine 800 is invoked. When subroutine 900 is invoked, string variable OUT$ carries the file output. (Data line 5 contains the total number of file lines. If you modify the program for fewer or more than 500 lines, change the data contents of line 5 accordingly.)

Following the main program listing are very simple examples of routines that invoke the subroutines at 800 and 900. These can be used to test the basic package of lines 0 to 991. One useful thing to note is that if you input nothing (i.e., hit return after typing nothing at all), the file entry thereby made consists of forty blanks. This can be used to erase previous entries. Second, note that *leading* blanks before the first nonblank character are stripped. Thus, all entries are left-justified. Third, a few punctuation characters, *particularly* the comma, cannot to be entered in the file (the comma is, used as a separator of data items, so a comma cannot appear within a data field). The use of such invalid characters will not damage the file, but the data won't be saved the way you meant it to be.

After using this program and appended applications, a SAVE command will tape the entire package, including all file entries, for future retrieval and data modification.

The Program

```
]0     REM DATA BASE PROGRAM
1      REM LINE 5 IS # OF ITEMS
5      DATA 500
10     DATA

11     DATA

12     DATA

13     DATA

14     DATA

15     DATA
•
•
•
500    DATA

501    DATA

502    DATA

503    DATA

504    DATA

505    DATA

506    DATA

507    DATA

508    DATA

509    DATA

510    :
511    :
798    GOTO 891
799    :
800    REM ROUTINE TO INPUT DATA
801    :
802    REM CALLING ROUTINE GIVES
803    REM FILE # AND INPUT STRING
805    :
806    RESTORE : READ XX
807    IF FM > XX THEN PRINT : FLASH : PRINT
       "CANNOT HAVE FILE NUMBER ABOVE
       ";XX;" !": PRINT : NORMAL : RETURN
808    IF FM < = 0 THEN PRINT "INVALID FILE
       #!": RETURN
810    REM COMPUTE MEMORY LOCATION
815    ML = FM * 46 + 2069
816    REM DATA LINE # = FM + 9.
818    :
820    REM PAD TO 40 CHARS.
825    IF LEN ( IN$ ) > 40 THEN FLASH : PRINT
       : PRINT "CANNOT ENTER;STRING TOO
       LONG!": PRINT : NORMAL : RETURN
830    IF LEN ( IN$ ) = 40 THEN GOTO 855
840    FOR I = 1 TO ( 40 — LEN ( IN$ )):IN$ =
       IN$ + " ": NEXT
850    :
851    REM POKE DATA INTO FILE
855    FOR J = ML TO ML + 39
857    IN$ = RIGHT$ ( IN$,( ML — J + 40 ) )
858    C1 = ASC ( IN$ )
860    POKE J,C1
867    NEXT
890    RETURN
891    :
899    GOTO 991
900    :
901    REM OUTPUT FROM FILE ROUTINE
902    OUT$ = CHR$ ( 0)
903    REM CALLING ROUTINE GIVES
904    REM DESIRED FILE # ( FM )
908    :
910    REM COMPUTE POINTER
911    RESTORE : READ XX: IF FM > XX THEN
       PRINT : FLASH : PRINT "REQUESTED FILE
       # EXCEEDS LARGEST FILE # WHICH IS
       ";XX;" !": NORMAL : RETURN
912    :
915    MP = FM * 46 + 2063
916    :
920    K1 = 16:K2 = 256:K3 = 4096
925    A1 = MP / K3:A = INT ( A1 )
927    B1 = A * K3:B2 = MP — B1
930    B = INT ( B2 / K2 )
935    C1 = B * K2:C = INT ( ( B2 — C1 ) / K1
       )
938    D1 = C * K1:D = B2 — C1 — D1
940    POKE 126,A * K1 + B
942    POKE 125,C * K1 + D
945    CF = PEEK ( MP + 6 )
946    REM BLANK FIELD?
947    IF CF = 32 THEN RETURN
948    READ OUT$
950    RETURN
991    :
```

Sample Application Routines

```
1000   REM A SIMPLE PROGRAM TO TEST
1001   REM INPUT CAPABILITY
1010   INPUT "ENTER TO FILE NUMBER...?";FM
1020   INPUT "STRING OF CHARACTERS TO BE
       INPUT ?";IN$
1030   GOSUB 800
1040   PRINT "ENTERED"
1050   STOP
1100   REM A SIMPLE PROGRAM TO TEST
1101   REM OUTPUT CAPABILITY.
1110   INPUT "RETRIEVE FROM FILE #.( 1 —
       500 )?";FM
1120   GOSUB 900
1130   PRINT "THE CONTENTS OF FILE #
       ";FM;" IS :": PRINT
1140   PRINT OUT$
1150   STOP
```

George V. Kinal, Washington, D.C.

A Definite Opinion

I recently canceled subscriptions to two other magazines, both for the same reasons. Both, while purporting to be for Apple, are heavily TRS-80 oriented. It is a pain to struggle to get a TRS-80 program running on the Apple, and it is most exasperating to read an ad for a promising program, get to the point of rushing out to buy it, and then find at the bottom "TRS-80, Level II Basic" instead of "Applesoft." Most of these programs are not available for the Apple II, nor apparently will be, because the authors refuse to write new 6502 code.

Therefore, I do not wish to be exposed to such frustration. I do not wish to buy a computer that has a reputation for system problems. Would you buy a program that habitually crashes and destroys your file when you get back in, though you've been working all day on it? Would you buy a computer that, after you work a straight seventy-two hours on a sophisticated program and try to RUN it, responds by zeroing out all the user memory and asking you for memory size? That's the TRS-80 for you! I've read a lot of this in a magazine after it changed from Apple to TRS-80 coverage halfway through my subscription!

Since the software house behind the magazine is dragging its heels on rewriting programs to run on the Apple II (eight for Apple, sixty-nine for TRS-80), I will not put up with that source of frustration.

I will stick to publications that deal with the Apple II. It has a user-accessible monitor, full memory support, recovery from system hang-ups without loss of code, customer support, and a crash-proof DOS. I would like any other computer—6502 or alien—to make that claim truthfully. The TRS-80 never gave me a break, and the Pet has given me some mulish problems (due to utter absence of a user monitor, reset button, etc.). But Apple has given me for two and a half years freeflowing, troublefree operation. You should see the fantastic programs I shaped or bought for it! One is uninterruptable, another uses lower case, etc.

P.S. I would like to correspond with anybody who has obtained a score of 350 on *Adventure*. I can only get 349.
Paul R. Wilson, Bergenfield, NJ

Hot for Hotline
I enjoyed the March Issue [Women in Microcomputing] of *Softalk* and am very interested in finding out the Hotline number for Apple users. The article never gave the number.

Thank you for this information.
Jane Clark, Loretto Academy, El Paso, TX

At present, Apple requests that you try asking questions of your dealer first; if your dealer doesn't know, there's a special hotline dealers can call. If you cannot find the answers that way, call Softalk *for the user Hotline number.*

Three Problems and a Bit of Praise
In your current issue of *Softalk* there are twenty-three photographs and five drawings of the Apple computer and not one of the photographs or sketches is an Apple III. On page 35, there is a full-page ad concerning Applefest '81 wherein they say, "See it all at Applefest"—and, guess what—a photograph of an Apple II.

Now, I'm not complaining about the Apple II, for I have two of them and they are fine machines—so say my employees. Based on their rave reviews I went out and bought myself an Apple III. That was my first mistake. The instruction manuals assume that you are an experienced programmer; in fact, they tell you that in the manual introduction. On the Apple III, it is necessary to program into the machine the number of disk drives you have and whether there is a printer. For three months now, the dealer has been trying to find out how to do it, but with no success. Several phone calls and a letter to the Apple people have gone unanswered. By using the emulation disk I can run the Apple III as an Apple II, but is it fair to the customer to have to pay more than $4,000 for a machine that is only doing the job of a $1,500 machine? Scuttlebutt out of Computerland has it that about three dozen Apple engineers have been fired in the last thirty days—could it be because of the Apple III or just scuttlebutt?

The only reference I saw to the Apple III in your magazine was a comment on page 52 that the Computer Case Co. was selling a heavy-duty box to put it in. Are they trying to tell me something? I'm beginning to have the feeling that I may have bought myself an Edsel but only time and Apple's integrity will tell.

I must agree with Mr. Velleman and Mr. Crawford (April Open Discussion) that the purchaser is entitled to know the limitations of a program, as they all have them. I don't see any dealers offering a ten or fifteen-day free trial to try their products—it's cash on the barrelhead and no guarantees.

I'd like to say again that you do have a fine magazine, one that I would be willing to pay for, but how about devoting a few pages a month to the beginner who isn't ready yet for Pascal or machine language?
Joffre M. Rothenberg, Santa Rosa, CA

1. See Exec Apple, page 4.
2. Blame it on pirates.
3. See page 71.

THE BASIC Solution

By Wm. V. R. Smith

The response to our first subroutine has been great, and the letters show an impressive amount of programming ability among *Softalk*'s readers. Keep it up!

This month's subroutine was created and sent in by Dr. Robert Suden of Portago, Michigan.

Suden's subroutine, named *Date Parser*, prompts the user to enter a date, which the routine calls DA$. Then it finds the month, day, and year values and assigns them to the variables M, D, and Y, respectively, which it measures against set values.

The program's purpose is to check all input dates for validity. If someone enters a thirteenth month or a thirty-second of June, the routine will catch the error and reprompt. It will not catch an erroneous, but valid, date, nor will it catch February 30, should it be given.

The calling program must pass the cursor vertical tab value V and the horizontal tab value H to allow the routine to position the cursor.

Whenever your programs require correct date values, this routine is a handy one to use. Even if you don't now have any programs that need it, keep the routine in your library; we'll refer to it in later articles. Hint: Eventually, all Basic Solution subroutines will be put together to create a very useful personal program.

Be sure to substitute values for V and H (line 7030); otherwise, you'll get an error message.

If you try to run the subroutine by itself, you'll get a return-without-gosub error. It is not intended to stand alone, but to be called, via GOSUB 7000, from within a host program. If you want to run it just to see it work, temporarily delete line 7150, insert V and H values in line 7030, and HOME before you run.

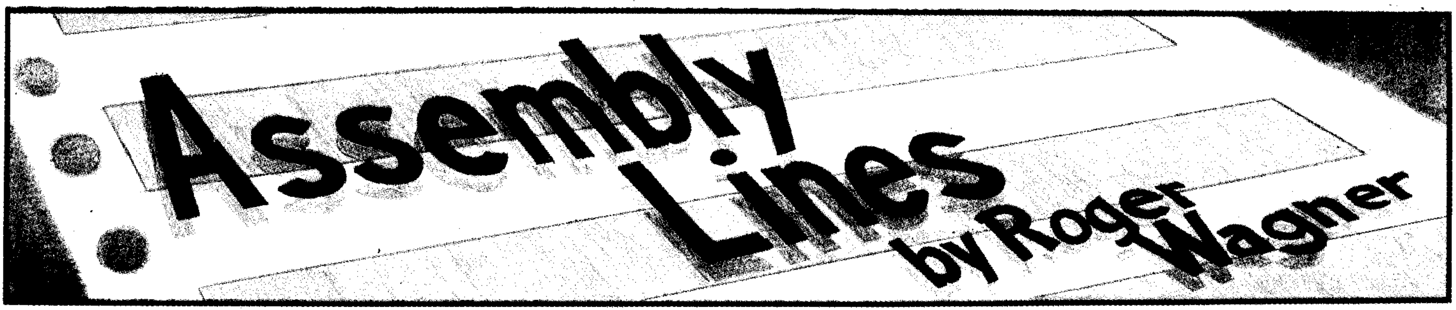
You may find you'd like the routine to call the attention of the user to the need to reinput in a stronger manner. If so, you can add:

```
7115 PRINT "[Control-G]"
7135 PRINT "[Control-G]"
```

This will ring the bell once. Of course, if you really want to make a racket, you can put in a whole bunch of Control-Gs before the closing quotes.

If you'd like to have a subroutine printed in Basic Solutions, send it to Softalk Basic Solutions, 11021 Magnolia Boulevard, North Hollywood, CA 91601. If it's published, you'll receive $10 credit from *Softalk* at your local computer store.

```
7000  REM  ****************************
7005  REM
7010  REM           DATE PARSER
7015  REM         DR. ROBERT SUDEN
7020  REM
7025  REM  ****************************
7030  VTAB V: HTAB H: INPUT "DATE MM/DD/YR";DA$
7040  FOR I = 1 TO LEN (DA$)
7050  IF MID$ (DA,I,1) < > "/" THEN NEXT I
7060  FOR N = I + 1 TO LEN (DA$)
7070  IF MID$ (DA$,N,1) < > "/" THEN NEXT N
7080  M$ = LEFT$ (DA$,I — 1):M = VAL (M$)
7090  D$ = MID$ (DA$,I + 1,N — I — 1):D = VAL (D$)
7100  Y$ = RIGHT$ (DA$, LEN (DA$) — N):Y = VAL (Y$)
7110  IF M > = 1 AND M < = 12 THEN 7130
7120  GOTO 7000
7130  IF D >=1 AND D < = 31 THEN 7150
7140  GOTO 7000
7150  RETURN
```



Everyone's Guide to Assembly Language, Part 9

Commands Covered So Far:

JMP	LDA	LDX	LDY	TAX
JSR	STA	STX	STY	TAY
RTS	INC	INX	INY	TXA
NOP	DEC	DEX	DEY	TYA
—	CMP	CPX	CPY	—
BEQ	BNE	BCC	BCS	—

Plus These Addressing Modes:

IMMEDIATE	ABSOLUTE
ZERO PAGE	IMPLICIT
RELATIVE	INDEXED

INDIRECT INDEXED
INDEXED INDIRECT

Figure 1

Part I: The Contest. With the usual comments about how hard it was to decide on a winner in mind, I hereby announce the winner of the contest as Steven Morris, of Queens, New York. His program combines a number of the principles we've discussed so far and also shows some nice touches in programming. It's an elegant use of all the codes given so far, and of particular interest is a self-modifying part wherein the program actually rewrites a small portion of itself upon user command.

I think it will be of interest, and also a good review, to go through Morris's listing to see what's been done. Before doing that, however, a little background on one more kind of tone routine is in order. This will make the program that much more understandable.

Last month I discussed simple tone routines in which the speaker was accessed at a constant rate for a given length of time. These two factors determined the pitch and duration of the tone played. A variation on this is to have the pitch decrease or increase as the tone is played, creating effects rather like the sound usually associated with a falling bomb or a rising level of something, respectively. This requires three variables, and without getting too technical, let me take a moment to illustrate with this chart:

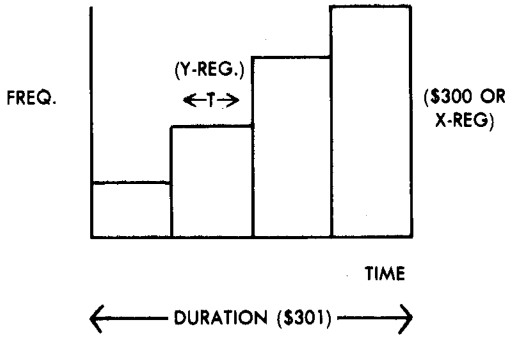


Figure 2

The vertical axis represents the frequency of the tone being played. Putting several tones together into a series over a period of time creates in this case a rising scale. As each tone is played, the pitch is increased. Each individual tone lasts some arbitrary time, *T*, and put together, the series lasts an overall time period, labeled here as *Duration.*

If the pitch is decreased by a certain amount each time, the pattern is reversed. This is sometimes called a *ramp* tone pattern. In parentheses, I have indicated how each of these values is determined in Morris's program.

Here is a source listing of the program:

```
*****************************
*      By Steven Morris . . .     *
*****************************
*
   OBJ $302
   ORG $302
*
PTCH EQU $300
DRTN EQU $301
SPKR EQU $C030
PREAD EQU $FB1E
PB0 EQU $C061
PB1 EQU $C062
GRSW EQU $C050
TXTSW EQU $C051
CLRSCR EQU $F832
*
LOOP DEX          ;  DEC THIS DELAY
   BNE CYCLE      ;  DONE? NO=SKIP CLK
*
CLK LDX PTCH      ;  REFRESH X-REG
   LDA SPKR       ;  CLK SPKR
```

```
* SPKR CLKS ONLY ONCE
* FOR EVERY ($300) PASSES
*
CYCLE DEY          ;  # OF CYCLE CTR.
  BNE LOOP         ;  DONE?
*                     NO=KEEP GOING
  DEC DRTN
  BEQ CHKPDL       ;  DONE W/ RAMP?
*                     YES=CHK PDLS
RAMP INC PTCH
  JMP LOOP
*
CHKPDL LDX #$00
  JSR PREAD        ;  READ PDL (0)
  STY PTCH         ;  SET DRTN
  INX
  JSR PREAD        ;  READ PDL (1)
  STY DRTN         ;  SET DRTN
  LDY #$7F
  CPY PB1          ;  #1 PRESSED?
  BCC TOGGLE       ;  BRCH IF YES
*
  INY            ;  #$7F—>#$80 ; AN EXCUSE
  TYA            ;  TO USE THESE
  TAX            ;  COMMANDS.
  CPX PB0        ;  #0 PRESSED?
  BCS LOOP       ;  BRCH IF NO
*
SCREEN JSR CLRSCR     ;  CLR TO BLK
  STA GRSW            ;  SHOW GRAPHICS MODE
  STA TXTSW           ;  SHOW TEXT MODE
  JMP SCREEN
*
SETDEC TAY         ;  USE UP THIS CODE
  LDX #$CE         ;  OPCODE FOR 'DEC'
  TXA
  CMP RAMP         ;  IS IT 'DEC' NOW?
  BEQ SETINC       ;  BRCH IF YES.
  STA RAMP         ;  NO. MAKE IT 'DEC'
  RTS
*
SETINC LDX #$EE    ;  OPCODE FOR 'INC'
STX RAMP
  RTS
*
TOGGLE JSR SETDEC
  JMP LOOP
*
```

This lists in memory as:

```
*300L

0300-  38             SEC
0301-  A5  CA         LDA   $CA
0303-  D0  06         BNE   $030B
0305-  AE  00  03     LDX   $0300
0308-  AD  30  C0     LDA   $C030
030B-  88             DEY
030C-  D0  F4         BNE   $0302
030E-  CE  01  03     DEC   $0301
0311-  F0  06         BEQ   $0319
0313-  CE  00  03     DEC   $0300
0316-  4C  02  03     JMP   $0302
0319-  A2  00         LDX   #$00
031B-  20  1E  FB     JSR   $FB1E
031E-  8C  00  03     STY   $0300
0321-  E8             INX
0322-  20  1E  FB     JSR   $FB1E
0325-  8C  01  03     STY   $0301
0328-  A0  7F         LDY   #$7F
032A-  CC  62  C0     CPY   $C062
032D-  90  27         BCC   $0356
032F-  C8             INY
0330-  98             TYA
0331-  AA             TAX
0332-  EC  61  C0     CPX   $C061
0335-  B0  CB         BCS   $0302
0337-  20  32  F8     JSR   $F832
033A-  8D  50  C0     STA   $C050
033D-  8D  51  C0     STA   $C051
0340-  4C  3A  03     JMP   $033A
```

```
0343-  A8          TAY
0344-  A2  CE      LDX   #$CE
0346-  8A          TXA
0347-  CD  13  03  CMP   $0313
034A-  F0  04      BEQ   $0350
034C-  8D  13  03  STA   $0313
034F-  60          RTS
0350-  A2  EE      LDX   #$EE
0352-  8E  13  03  STX   $0313
0355-  60          RTS
0356-  20  43  03  JSR   $0343
0359-  4C  02  03  JMP   $0302
```

I'll try to explain each part of the program, hopefully with a proper balance of enough detail to jog your memory and enough brevity to keep things reasonably short.

If all this seems overwhelming, you're trying to read through it too fast. Go back through it slowly, taking your time. Have a nice cup of tea while you're at it.

Remember, we're packing eight issues worth of subject matter into one program. Don't worry if the fine details of the tone routine escape you. The important part is to make sure that you at least recall the existence and general nature of each individual command used in the program.

To explain the program, the easiest place to start is actually at CHKPDL, where the paddles are checked for new values at the end of each ramp series (line #35 @ $319). The X register is loaded with a $00 to tell the computer we want to read paddle #0 in the next step, then JSR to $FB1E. That returns with the Y register holding the value of the paddle ($00 to $FF), which is then stored in location $300, labeled "PTCH" (for pitch). The X register value is then incremented from $00 to $01 on line #38, and paddle #1 read and stored at $301 for the duration value.

If paddle button #1 is pressed, location $C062 will hold a number greater than $7F. To check for this the Y register is loaded with $7F and compared against $C062. If $C062 holds a value greater than $7F, the branch carry clear (BCC) will be taken (Y register < memory location = carry clear). We'll see what that does later.

If the value is less than $7F, program execution will fall through to line #45. Here the $7F is increased to $80 and that value passed to the X register via the accumulator. These steps are here to exercise the INY, TYA, TAX commands, and to allow us to use the CPX command next to fulfill the contest requirements. At line #48 the comparison is done. If the X register is greater (remember it holds a $80 here), the button is not pressed and the branch carry set will be taken (X register > memory loc = carry set) that sends us to the main tone loop.

At entry to this loop, the X and Y registers hold rather arbitrary values, but the overall theory is that, starting at CLK on line #21, the X register is loaded with the pitch value and the speaker clicked once. At #26 the Y register is decremented; this is a counter for the length of that pitch value. Jumping back to loop, the net effect is that the program will make *n* passes through before clicking the speaker once, where *n* is the pitch value held in $300. This creates the delay between clicks needed for a given tone.

Judging the contest was a snap.

Southwestern Data Systems photo.

The length of that particular tone is determined by the Y register. When it reaches a value of $00, the BNE (branch not equal) fails and the counter for the overall duration is decremented. As long as there's time left (that is, DRTN > $00), the next test fails (BEQ = branch if equal to zero) and the pitch value is incremented.

Going back to loop plays this next note until all the notes in the series have been played. Incrementing pitch gives a descending note pattern. (Recall that the greater the pitch value, the lower the tone played.)

When DRTN does reach zero, the program branches to the paddle check routine that we started in. Let's see what happens when a button is pressed.

If button #1 is pressed, the program goes via toggle to SETDEC. This clever section (ignore the TAY) loads the X register with the value $CE. This is the opcode for DEC (decrement a memory location), and checks to see if the part of the program at ramp ($313) holds a $CE. (Notice that it does in fact start out as an $EE, the opcode for INC (increment a memory location).

If the comparison fails, that is, there is not a $CE currently there, the $CE is stored at ramp, the RTS (return from subroutine) returns to toggle and the JMP loop sends everything back into the tone loop, this time with a DEC PTCH there instead. This gives an ascending pitch series.

If the comparison is true, it means that a $CE was put there earlier, and the BEQ goes to SETINC, which restores the code for INC at ramp ($313), and then returns with the RTS, JMP loop as in the previous case.

These two options give the program the ability to rewrite itself, an interesting and powerful idea.

If paddle button #0 is pressed, the branch at line #49 fails and the program falls into an infinite loop at screen ($337). In this loop, the screen is cleared to the color black by the monitor routine at $F832.

Locations $C050 and $C051 are softswitches mentioned in earlier articles. Remember that accessing these changes the display mode of the Apple. The screen can be viewed either in a text mode or a graphics mode. Accessing $C050 on line #52 sets the graphics mode, so the screen appears black. Accessing $C051 sets the display to text, which appears as inverse "@" signs.

The JMP screen repeats this cycle back and forth so fast that you don't actually see the flicker, just an interesting pattern created by the screens switching faster than your monitor screen can display them.

At this point you have to hit reset to end.

There were a number of other excellent entries. Honorable mention should be made of Steve Hawley, Ray Ransom, Stephen Gagola, Jr., and Matt Brookover for their efforts.

Part II: The Stack. One of the more obscure parts of the operation of the Apple is related to something called *the stack.* This is a part of memory reserved for holding return addresses for gosubs and for-next loops, and a few other operations in direct machine code.

If you want to impress your friends with your knowledge of machine language, just throw this term around in a confident manner and they'll figure you must be an expert!

The stack is a reserved part of memory from $100 to $1FF. The 256 bytes (a page of memory, remember . . .) here are reserved for storing temporary values during the operation of the computer.

The stack can be thought of like those spring-loaded plate holders they have in restaurants. Plates are loaded onto the top of a cylinder with a spring-loaded platform in it. As more plates are added, the bottom one gets pushed down. The plates must always be removed in the opposite order from that in which they are put in. The catch phrase for this is LIFO, for last-in, first-out. The first location loaded in the 6502 stack is $1FF. Rather than push everything down toward $100 each time a new value is put on the stack, the 6502 has a stack pointer that is adjusted as new data is added. Successive values are added in descending order, with the stack pointer being reset each time to indicate the position of the last value put in. Thus the table is created in a reverse order, building downward.

The technical details of its operation are not required to make good use of it, though. One of the most convenient things the stack can be used for is to hold values temporarily while you're doing something else. Normally, in a program, we'd have to assign a zero page location to hold a value. For instance, consider this program:

```
 1 ********************************
 2 *    BYTE DISPLAY PROG. #1     *
 3 ********************************
 4 *
 5   OBJ $300
 6   ORG $300
 7 *
 8 CHR EQU $06
 9 PRBYTE EQU $FDDA
10 COUT EQU $FDED
11 PREAD EQU $FB1E
12 HOME EQU $FC58
13 *
14 START JSR HOME
15 GETCHR LDX #$00
16   JSR PREAD
17   STY CHR
18   TYA
19   JSR PRBYTE
20   LDA #$A0  ;  SPACE
21   JSR COUT
22   LDA CHR
23   JSR COUT
24   LDA #$8D  ;  RETURN
25   JSR COUT
26   JMP GETCHR
```

This will be listed by the monitor as:

```
*300L

0300-   20 58 FC     JSR  $FC58
0303-   A2 00        LDX  #$00
0305-   20 1E FB     JSR  $FB1E
0308-   84 06        STY  $06
030A-   98           TYA
030B-   20 DA FD     JSR  $FDDA
030E-   A9 A0        LDA  #$A0
```

```
0310-   20 ED FD    JSR   $FDED
0313-   A5 06       LDA   $06
0315-   20 ED FD    JSR   $FDED
0318-   A9 8D       LDA   #$8D
031A-   20 ED FD    JSR   $FDED
031D-   4C 03 03    JMP   $0303
```

This program gets a value from $00 to $FF from paddle #0, and stores it in location $06. This is needed because the JSR to $FDDA (a handy routine that prints the hex number in the accumulator) scrambles the accumulator and Y register. We want to keep the value at hand because the ASCII character corresponding to it is then printed out right after the number using COUT. The cycle then repeats until you press reset.

Location $06 is used for only a moment each pass to temporarily store the value. In addition, it commits that zero page location to use and thus limits our choices when we need other ones to use. A better system is to make use of the stack. The commands to do this are PHA and PLA. PHA stands for push accumulator onto stack. When this is used in line #17, the value currently in the accumulator is put onto the stack. The accumulator itself goes unaltered, and none of the status flags such as the carry or zero flags are conditioned. The value is simply copied and stored for us.

Later on, when we want to retrieve the value, the PLA on line #21 (stands for pull accumulator from stack) pulls the value back off the stack into the accumulator. A PLA command does condition the zero flag (and also the sign bit, which I've not covered yet).

Important: For each "PHA" there must be a "PLA" executed before encountering the next "RTS" in a program.

Here's the revised program:

```
 1 ******************************
 2 *     BYTE DISPLAY PROG. #2   *
 3 ******************************
 4 *
 5    OBJ $300
 6    ORG $300
 7 *
 8 PRBYTE EQU $FDDA
 9 COUT EQU $FDED
10 PREAD EQU $FB1E
11 HOME EQU $FC58
12 *
13 START JSR HOME
14 GETCHR LDX #$00
15    JSR PREAD
16    TYA
17    PHA
18    JSR PRBYTE
19    LDA #$A0 ; SPACE
20    JSR COUT
21    PLA
22    JSR COUT
23    LDA #$8D ; RETURN
24    JSR COUT
25    JMP GETCHR
```

This will list like so:

```
*300L

0300-   20 58 FC    JSR   $FC58
0303-   A2 00       LDX   #$00
0305-   20 1E FB    JSR   $FB1E
0308-   98          TYA
0309-   48          PHA
030A-   20 DA FD    JSR   $FDDA
030D-   A9 A0       LDA   #$A0
030F-   68          PLA
0310-   20 ED FD    JSR   $FDED
0313-   A9 8D       LDA   #$8D
0315-   20 ED FD    JSR   $FDED
0318-   4C 03 03    JMP   $0303
```

The stack is also used automatically by the 6502 for storing the return address for each JSR as it's encountered. Each time you do a PHA, this address is buried one level deeper. You must have done an equivalent number of PLAs at some point in the routine before reaching the next RTS to have things work properly.

Also remember, if you want to store more than one value, you must retrieve the values in the opposite order in which they were stored. Once a value is removed from the stack with a PLA, it is essentially gone forever from the stack unless you put it back directly.

There is a limit to how much you can put in the stack. The limit of sixteen GOSUBs and FOR-NEXT loops in Basic is related to the use of this. Technically you can put 256 one-byte values, or 128 RTS addresses on the stack, but the Apple also uses it for its own operations, and, many times, you have Basic going, too.

In general, though, it rarely fills up unless you're getting extreme in its use, and at that point the code is probably so confusing in nested subroutines that you may want to consider a rewrite anyway!

Try putting this in some of your own programs; I think you'll find it quite useful. Until next month then, Happy Appling!

Errata: Please note that in the first disassembly, given on page 67 last month, the second line should have read:

```
0303-   4C 00 03    JMP   $0300
```

FAIR DAYS

Softalk photos

Left to right: Roger Wagner (standing) and Tom Burns man the Southwestern Data Systems booth. Apple's absence was eased by the unofficial presence of several Apple execs including Stephen Shank, far east business manager. Two awards were given at the show. Steve Gibson of California Pacific accepts Softalk's Most Popular Program Ever award for Super Invader. Tim Smith demos his Olympic Decathlon after receiving a programmers award from Creative Computing. Smith's publisher, Microsoft, was represented by Vern Raburn, Consumer Products division prexy, and Bill Gates topper of the parent company.

Left to right: Some traveled long distances to go to the Faire. Jerry Vandiver, coauthor of Vanlove's 1981 Apple Software Directory, was on the last leg of a motorhome trip around the country when he arrived in San Francisco. Harry Harper came up from Down Under to demo a snazzy new 80-column board for the Apple. Wes Thomas, publisher of VisiNews, a VisiCalc newsletter, was in from New York. Another journalist present was Ben Rosen, publisher of Rosen's Electronics Newsletter. Retailers roamed the floor looking for new products, including Jim Sadlier of Southern California's Computerland of Lawndale. Robert Schoenburg, president of Continental Software, and Bill Baker, president of Information Unlimited, were among exhibitors.

Left to right: Milo Street of Street Electronics displayed the Echo II, which made the Apple talk. Unique among the booths was this one from Adventure International in a bamboo construction. This sly-looking fellow has beguiled adventure fans for the past eighteen months. He's Scott Adams, architect of the ten Adventure International fantasies. Robert Woodhead, getting ready to launch a Dungeons and Dragons program for his Sir-Tech firm, was in evidence checking out the marketplace.

Left to right: Lunch was a catch-as-catch-can affair for exhibitors. Softsel's Dave Wagman, Craig Moody, and Bob Leff grab a snack during a lull in the action. Lynne and Jeff Mazur of West-Side Electronics showed a new clock for the Apple and passed out apple-flavored jelly beans to passersby. What are all these people looking up to? It's the new arcade game from Nasir, Pulsar II, which was mounted in the top corner of the Sirius Software booth.

Left to right: Gary and Doug Carlston of Broderbund Software, who published Alien Rain, which led the Top Thirty poll for three months. Stan Goldberg, director of Micro Lab, exhibited through the Software Express booth and was a catalyst in the organizational meeting of the Software Publishers Association. The Syntauri music keyboard for the Apple drew some unusual users. The dragon actually is Charlie Kellner of Apple, codesigner of the Syntauri. The other gentleman is Herbie Hancock, more often associated with jazz fests as opposed to computer fairs. Behind him sits a defrocked (descaled?) Kellner.

Left to right: There was never a surcease of activity at the On-Line Systems booth. In the foreground is Ken Williams, while Roberta can be spotted in the center, rear. Gary Atkinson and Walt Wilson demoed RH Electronics language card and their new fan. Not all the attendees found every facet of the doings fascinating. But at least some folks went home with the equipment to while away their leisure hours the way most Softalk readers do—with an Apple.

Birds migrate north in the spring and congressmen junket in the summer. Microcomputer firms likewise do their own thing—they exhibit—and their schedule this year looked something like that of the birds.

Omitting those soirees that had little or no impact on the Apple market, there were the West Coast Computer Faire in San Francisco in April, the National Computer Conference in Chicago in May and Applefest in Boston in June. And for many companies, there was also Apple Expo, a March-April event conducted in Dallas, New York, Chicago, and Los Angeles.

Each of these events has a flavor all of its own. The West Coast Computer Faire focuses on microcomputing in an all-encompassing way. Exhibitors supported many breeds of personal computers and attendees could get a fair sampling of each. General emphasis is on the end-user attendees.

NCC has more the flavor of a trade show, although thousands of end users and potential end users also find their way onto the exhibit floor. NCC is the biggie, in terms of floor space, exhibitors, attendees, and the kinds of computing things you'll find there. IBM, Honeywell, Xerox Data Systems, Digital Equipment Corporation, and hundreds of others exhibit there. The micro folk are there, but they aren't as dominant as they are at the West Coast Computer Faire.

Applefest, probably being conducted as you read this (June 6 and June 7) shares with Apple Expo the distinction of being an Apple specific show. But again, similarities fade at that point.

Applefest primarily serves the end user or potential end user, with approximately one hundred exhibitors, plus seminars on various facets of the use and capability of the Apple.

On the other hand, although Apple Expo, sponsored by Apple Computer Inc., had a segment open to the general public, its main focus was on the dealer and potential dealer.

All but one-half of one day were given over to product demonstrations and seminars designed to aid the Apple franchisee or would-be franchisee. More than thirty suppliers of Apple-related products accepted invitations to be a part of the Expo.

These pages highlight activities and people who were at the West Coast Computer Faire and Apple Expo.

So Late at the Faire. The West Coast Computer Faire has an honored place in Appledom because it was there, in 1978, that the first widespread excitement over the Apple was generated. So it was with some disappointment that die-hard Apple enthusiasts searched in vain for this year's Apple booth. Apple had a conflicting appointment with its own Expo and wasn't able to make the show.

But that initial disappointment faded rapidly at the sight of the many booths dedicated to Apple products.

In the eyes of Jim Warren, Faire coordinator, this was the year that marked all-around maturation. The marketplace seems to have matured, the exhibitors had matured, and the staff of the Faire had matured in its ability to handle the crowds.

And crowds there were. The conference program included 164 speakers and there were more than 520 exhibits. San Francisco Civic Auditorium was site of the three-day affair and it was packed both Friday and Saturday, with a little letup on Sunday. In all, 31,754 persons attended.

Clearly, the most dramatic single event of the Faire was the unveiling of the Osborne I, Adam Osborne's entry into the field of personal microcomputers. Some of the suspense was undercut by a *Wall Street Journal* article days before the Faire that gave complete technical and pricing details for the machine, but it was still an object of great interest and curiosity.

Nothing else could approach the curiosity generated by a new microcomputer, but signs of steady growth were everywhere.

Last year's hi-res graphics spectaculars were taken for granted this year.

There were two type-ahead buffers to complement Apple's word processing programs. Last year there were none.

There was a friction feed platen for the Epson printer. Last year, Epson was not yet a factor in the marketplace.

Last year, Microsoft's *Adventure* and Scott Adams's adventures were it for the Apple. Now there are several, including many in hi-res. Last year's adventures required two-word syntax; one of this year's models can take full, compound sentences.

Eighty columns on the Apple CRT was brand new last year; this year there were a handful, including an invader from Australia.

Last year, the warrior in the *Temple of Apshai* was state of the art for Apple animation. This year, *Olympic Decathlon* set a new standard for animating the human figure, while *Raster Blaster* accomplished the same thing for objects.

The striking contrast in exhibit space brought home the fact that microcomputing is still close to its cottage industry roots—the earnings of Apple, Tandy, Warner Communications, and Commodore notwithstanding.

While companies like Adventure International and Programma were taking up several regular-sized booth spaces, the one- or two-person companies and the hobbyists showed their wares in minibooths on the outer extremities of the exhibit floors and in hallways. And, as is often the case in this young industry, it wasn't always the exhibitor with the most pizzazz and show who had the hot new item.

The seminars held in conjunction with the exhibits varied considerably in quality and general interest. But most seemed well attended by cognoscenti who had pertinent questions and contributions.

Liveliest of the seminars was probably the one on legal safeguards and copyrights, a relatively dry and innocent sounding title. It started that way, with San Francisco attorney David B. Harrison recounting the law of software copyright.

It was when Ron Williams of Apple, standing in for Apple's Mike Kane, took the podium that things started to liven up.

Williams directly addressed the issue of software piracy, isolating two breeds of pirate—the organized thief who will change a program and sell it as his own and the casual copier who makes a few copies for friends. Williams defended Apple's copy protect policy and explained that software authors needed their work protected or they might defect from the industry.

This brought several rebuttals from the audience, including one attendee who openly admitted to copying disks and vowed to continue the practice regardless of law or ethics. An impromptu open discussion session ensued.

The atmosphere was so heated that the third speaker, Dr. Mark Spohr of Medsoft, looked at the audience and said, "It's like entering a roomful of vipers. You walk through and you have to trust."

So animated was the discussion that attendees pursued various aspects of the problem in the hallway after the close of the seminar. So much for the dry sounding seminars.

The Apple Roadshow. Apple's Expo also had seminars, but these served an instructional purpose—covering such subjects as how to hire skilled employees, how to improve service and maintenance, effective marketing techniques, and so forth. No subject was approached as emotionally as the piracy issue at the West Coast Computer Faire, although gibes by dealers against those of their ilk who practiced discounting sales techniques were favorably received.

Exhibitors had a chance for nearly three days to go one-on-one with dealers to explain what their products were about and how to market them properly. It's not suprising, then, to learn

GOTO 62

VENTURES WITH VISICALC

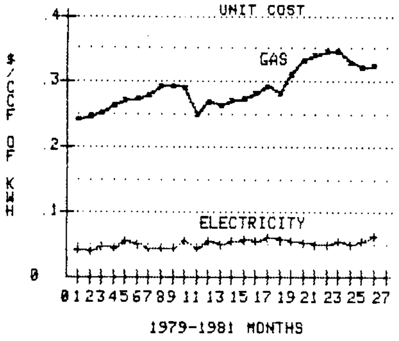
BY FRANK MALONE

A number of readers have responded to our call for VisiCalc *applications. One of the most detailed applications is the subject of this month's Ventures with VisiCalc. The guest columnist is Otto K. Wetzel, Jr., of Dallas, Texas.*

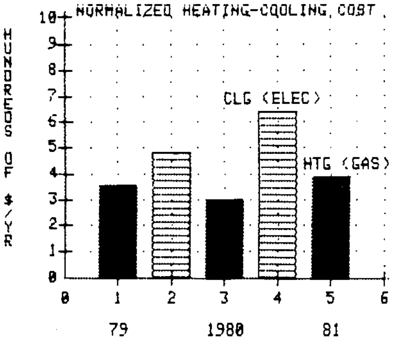
Have you ever wondered—when you received your apparently ever-increasing utility bills—about the ability to correlate your costs with the weather? This thought has become more and more intriguing, particularly as energy has become increasingly expensive since 1973. A recently developed *VisiCalc* program does the job.

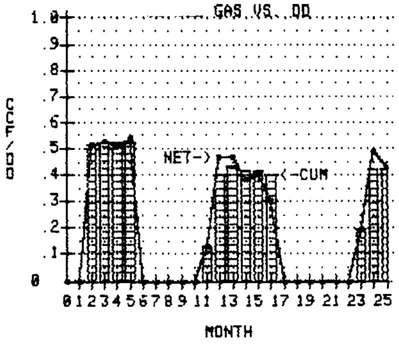
To start, you need, of course, weather data. For the princely sum of $3.30 per year, you can get, from the National Climatic Center, U.S. Department of Commerce—NOAA, Federal Building, Asheville, NC 28801, a monthly summary (which comes about two months late) of the weather information in your immediate area. Among other things, this information gives the the high and low temperatures and the heating and cooling degree days.

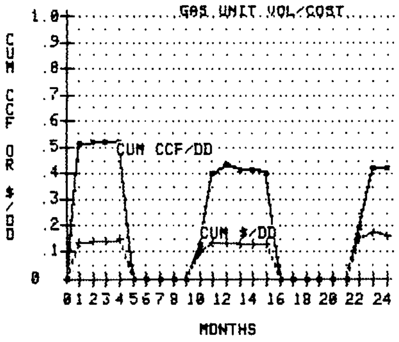
A degree day is a measure of the heating or cooling requirements against a base of 65 degrees Fahrenheit. The theory behind a degree day is that any building generates a certain amount of its internal heat from people, lights, machinery, and so on. If the outside temperature is 65, these other sources of heat will keep the building somewhere in the middle of the comfort zone, which is usually defined at 68 to 78. Thus, if the temperature during a day averages 60, in weather parlance this is called a "five-degree heating day." Conversely, if the average temperature goes up to, say, an average of 75, then this would be called a ten-degree cooling day. Now, in the days of higher energy costs and, thus, more concern about keeping thermostats set high or low, this rule is not as reliable as in the past, but it remains what is reported, and we'll use it in our correlations.

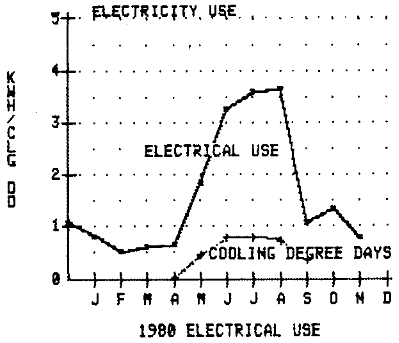


Basically, you take the degree-day information, beginning at a given point (in the example, it was January 1979) and tabulate this, day-by-day, for each month. It is simple, of course, to add months to the file whenever you receive new information. To aid in picking off particular temperatures for particular days, label the rows and columns identically to the way they appear on *VisiCalc*. Thus, for example, the data for January 4, 1979, would appear on the *VisiCalc* screen at row B24, and would be so labeled in the printout. On the same sheet, below this information, tabulate the actual heating years (July 1 to June 30) and cooling years (January 1 to December 31),









thus showing them against the normal for the arêa. Doing this will enable you to normalize your correlations.

The next step is to tabulate the pertinent information from your utility bills. In the sample home, there is gas heating and electric air conditioning. You simply tabulate this information through the date the meter is read, entering the date, the amount in hundreds of cubic feet or kilowatt hours, and the cost.

The next step is the only tedious step in the whole operation. Because the meter is not read on the first of the month, to get a true correlation against degree days you must sum up the degree days between meter readings. To do so, adopt a custom with respect to the billing date. In this example, it's assumed that the day on which the meter is read is included in the degree days for the month billed. Then it is a simple mathematical operation to accumulate the degree days between billings. To do this, go back to the top table—which you've previously printed out—and note the appropriate column and row for the date of the last bill and the date of the current bill.

By having *VisiCalc* sum up the days after the previous month's billing date to the end of that month and adding that sum to the days through the date of the billing, you càn arrive at the degree days actually covered by the billing period. This is a bit tedious but, once done, never has to be touched again. Proof it with somebody else to make sure you haven't made an error.

The next step is to develop the various correlations. To begin with, you'll want to know the unit cost of energy, particularly if it is escalating rapidly in your area. This is a simple division and gives you a price in dollars—per hundred cubic feet of gas or per kilowatt hour of electricity. The figures for the sample home reflect a substantial increase over the period.

Next, you must make a judgment on the amount of energy used for other purposes. The sample home uses a gas water heater and a cooking stove with pilot lights in it. Thus, it is fairly simple to get an average of the amount of gas used during the summer for those two services. You'll also know the approximate time of the year when you turn off the furnace and turn it on. In those months in which you know the furnance has not been operated, merely subtract out the total bill and use an average for those two services, notably hot water heating and stove pilots, for the balance of the year. By subtracting this out, you get a net amount used for heating. Do the same for electricity.

Once you have this information, it is a simple matter to develop a correlation of degree days divided into the gas or, in the case of electricity, the net power used. Do this by month and on a cumulative basis, both of which are extremely simple in *VisiCalc*. Doing it on both a monthly and a cumulative basis enables you to analyze any abnormalities. Also, by cumulating, you take care of the shoulder months where you may or may not use the furnace or air conditioning because of the low number of degree days.

Then plot any of the derived information (in the sample, we used the *Apple Plot* program), which makes this kind of data much more meaningful. Examine the sample house plots to see how this works.

Of what practical use is this information? First, it can make you much more aware of the costs of heating or cooling the house when it is not occupied. Based on the early correlations in the sample home, energy consumption has been reduced by 25 percent since 1973, by installing storm windows, closing off unneeded rooms, and turning off the furnace or air conditioning when no one is home. We are now in the process of zoning the sample house. All these measures result in a reduction in what the utility costs would have been.

A secondary reason for all this information is that it enables study of solar applications, particularly through providing hard data for analyzing other heating and/or cooling situations. It is only a simple step from these correlations to project forward into future costs. With this information, assuming energy costs accelerate more rapidly than other costs, as it appears they will, you'll be in a position to judge if and when solar heating or cooling is economical for you. ⊐■

At 3.3, VisiCalc Spawns a Family

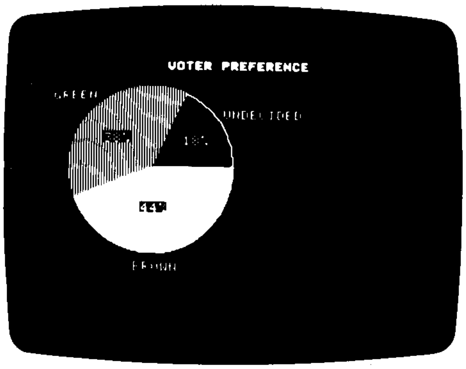
Spring is beloved by those of poetic spirit as a time of rebirth, a time when flowers return to the northern climes, trees sprout greenery, birds return from the south, and the weather has a beguiling mildness.

The season this year may also become heralded as a benchmark in the history of Personal Software and personal computers. The long silence and nearly ominous drought of new product from the publishers of *VisiCalc* has been broken with a spate of new software that almost resembles the breaking of a logjam during spring thaw on the Klamath River.

First came *Rainbow Writer,* followed by *Zork,* which zoomed to fourth on the Top Thirty list in its first month of general distribution, and an upgraded version of *Desktop Plan.*

But primarily the season will be noted for the introduction of an upgraded *VisiCalc* compatible with the 3.3 disk operating system and for the various kith and kin that have tagged along behind.

Father Calc and His Clan. *VisiCalc* is no longer a lone warrior doing battle against the forces of tedious and time-consuming hand calculations. Now he's the patriarch of a sizable family dedicated to that warfare. *Calc*'s kin are *Trend, Plot, Dex,* and *Term*; together they form what appears to be the most comprehensive and powerful family of software this side of the IBM 370.



All the packages understand the data interchange format (DIF) method of storing data used by Father *VisiCalc.* While each of the programs could probably stand on its merits alone, access to the power of *VisiCalc* gives each value beyond its intrinsic worth.

The summary explanations of the products that follow were generally culled from information provided by Personal Software and verified by brief tests of the software. Detailed looks at the abilities of each package will be found in the July issue.

Improving on the Best. There's a whole body of *VisiCalc* users out there who will be stunned to find that the program could be improved upon. By far the bestselling program for personal computers, *VisiCalc* has caused an entire subindustry to grow around it, from user groups and newsletters to companies providing utilities to ease use of the program.

The most salient change to *VisiCalc* is its adaptability to the 3.3 operating environment, the one enhancement that new Apple owners have been crying for over the past six months.

But more telling in the long run are the additions of Boolean functions which technically raise *VisiCalc* out of the category of the powerful electronic calculator and into some higher order of machine.

The truly awesome strength of the program has always been more in its latent abilities—there to be found and exploited by the innovative user—than in its more obvious abilities. The added Boolean functions give Father *Calc* even more latent power. For every user who adopts *VisiCalc* 3.3 for a mundane purpose there will be another who discovers a totally unthought of use for the program and its newly found abilities.

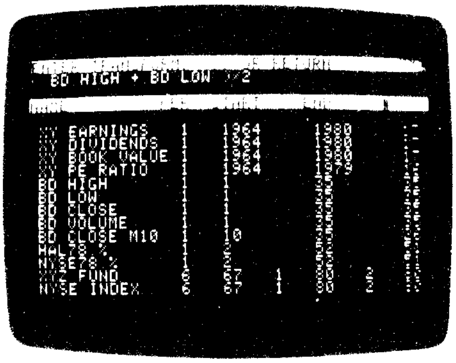
Also new to *VisiCalc* is recognition of relational mathematical symbols such as greater than, not equal to, etc., and added commands, such as edit, which allows the user to change a formula without rekeying the whole function.

How To Plot a Trend. *VisiPlot* and *VisiTrend* are both products of Mitchell Kapor, Micro Finance Systems. *Plot* is sold as a separate package, while *Trend* comes equipped with *Plot* in every instance.

The purpose of *VisiPlot,* of course, is to present whatever data you want to study in graphics form. A veritable plethora of output options exist: line charts, bar charts, area charts, pie charts, hi-lo charts, scatter charts, and combinations thereof.

Ease of use and versatility of titling features show Kapor's concern for the end user.

Plot can hold as many as sixteen data series as long as their combined number of data points does not exceed six hundred forty-five. Any individual chart can plot up to one hundred fifty points.

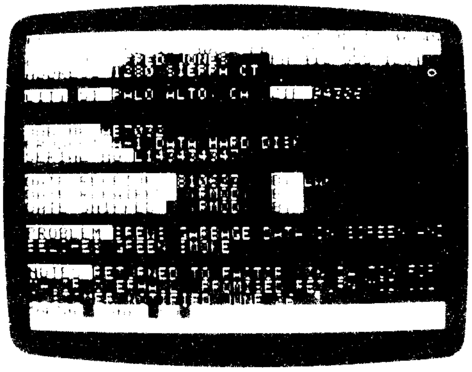


```
 BD HIGH + BD LOW /2

XY EARNINGS    1   1964      1980
XY DIVIDENDS   1   1964      1980
XY BOOK VALUE  1   1964      1980
XY PE RATIO    1   1964      1979
BD HIGH        1   1
BD LOW         1   1
BD CLOSE       1   1
BD VOLUME      1   1
BD CLOSE M10   1   10
HAL78 %        1
NYSE78 %       1
XYZ FUND       6   67   1           2
NYSE INDEX     6   67   1           2
```

No Trend for the Unwashed. *VisiTrend* fleshes out Father *Calc*'s family with statistical analysis abilities. *Trend* will do all manner of analysis that desktop computer owners have been wanting to do. Generally speaking, Apple owners will fall into three categories relative to *Trend*: those who don't know mean from mode from average should forget it—this is a sophisticated package that will leave the statistically unwashed in the lurch; those who will buy it because they think they should have it, even though the closest they come to statistical analysis is figuring batting averages; and finally, those who need trendline forecasting, Durbin-Watson statistics, linear multiple regressions, etc. This last group is likely to be jostling for position to exhaust their local retailer's stock.

Appending *Plot* to *Trend* gives added value to *Trend*'s output by providing graphic detail for *Trend*'s analysis.



```
          FRED JONES
          1380 SIEPPH CT
          PALO ALTO, CA         94305

          E7072
          1 DATA HARD DISK
          L14343437

              810627
              PMOD
              PMOD

PROBLEM   SEEMS GARBAGE DATA ON SCREEN AND
          SPEED IMOPE

NOTE      RETURNED TO FACTORY
          PROMISED
```

Father Calc's Weak Sister? At first glance, *Dex* looks like the weak sister amongst Father *Calc*'s kin. Even its manual modestly proclaims it to be an electronic index card, a term almost anachronistic to Apple owners.

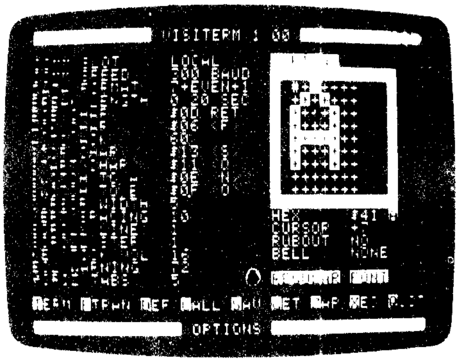
But the program, written by Peter Jennings, one of the founders of Personal Software, seems to have a depth that belies surface appearance. Fundamental to that depth is *Dex*'s cross-referencing ability.

Suppose you were to make out a *VisiDex* card titled pianists, and then you cross-referenced the subject with the names Horowitz, Ferrante & Teicher, Elton John, and Jerry Lee Lewis. Later, when you ask the program for all the data on Horowitz, the pianist card will be pulled, showing all the names you cross-referenced on it and proving that you're a person of uncommonly universal musical tastes.

Clearly this is an index card cut from a different stock of electronic cardboard.

The program can be used in entirely unformatted style or templates can be made for data that is more or less organized the same each time. It also knows the correct calendar displays for every month of the twentieth century. The calendar function is handy for keeping track of appointments, commitments, etc.

Dex is the one member of Father *Calc*'s family that stands alone, which figures, because you can neither plot nor statistically analyze your appointments calendar, nor are Boolean functions particularly germane to a list of pianists.



```
                 VISITERM 1.00
               LOCAL
               300 BAUD
               +EVEN+1
               0 20 SEC
               $0D RET
               $0C FF

                          HEX      $41
                          CURSOR
                          RUBOUT   NO
                          BELL     NONE

                    OPTIONS
```

Term Will Do the Talking. *Term* is the translator and the communicator for Father *Calc*'s clan. *Term*, written by Tom Keith, can talk to the big guys from IBM or to another Apple with equal ease.

The essence of the program is making the Apple an intelligent terminal so you can address the various computer libraries of information, access mainframe data processing capabilities, or just talk via Apple to your Aunt Vi in Victorville. Baud rates up to 1200 are accommodated.

The software features hi-res character text fonts in proportional spacing and a font editor that permits you to design your own display fonts. A provision for macros also eases use.

Father *Calc* and Cousin *Plot* hit your local retailer's door about the first of May. *Trend, Dex*, and *Term* await final documentation at this writing, but are either at your retailer as you read this or soon will be.

VisiCalc will run on any Apple II with DOS 3.3, 48K, and one disk drive. $199.95.

VisiPlot will run on any Apple II with DOS 3.3, Applesoft, 48K, and two disk drives. $179.95. The *VisiTrend/VisiPlot* package has the same requirements. $259.95.

VisiDex will run on any Apple II with DOS 3.3, 48K, and one disk drive. $199.95. *VisiTerm* has the same requirements as VisiDex, $149.95.

Poem

Horses race like wind
Running so sleek and pretty
Much like birds in flight.
—Leslie Evans

Little Speckled Mule Deer Fawn

I want to get to the other side of that shiny thing that my mother called a creek when she was alive. I would like some of that luscious looking grass over there.
—Leslie Evans

Lincoln School, Boise, Idaho, Photo

An Apple for Leslie

BY MELISSA MILICH

Imagine not being able to talk. Forget about sign language—you've only got the use of one finger. How would you express your feelings of joy, anger, or even hunger and pain?

Now imagine what it would be like to suddenly have the gift to communicate returned after two years. If you have a good imagination, you might begin to feel what life has been like for eighteen-year-old Leslie Evans.

Two years ago, Leslie, an athletic, dating, typical high school student and an avid equestrian since she was eight, fell from her horse. The accident caused brain stem damage that left her unable to walk, talk, or use sign language. But, despite the physical devastation Leslie's injuries brought about, her cognitive abilities were untouched.

Leslie had an active mind trapped in a very crippled body, and her only way of communicating was through a lap board with the alphabet stenciled on it. With only limited use in her right hand, it took all her strength to point to one letter at a time. A simple request, such as the television show she wanted to watch that night, took Leslie at least thirty minutes to spell out—provided her mother or another person was there to interpret. Anything more complicated was impossible since the slow process would tire Leslie and her mother long before it was completed.

But fourteen months ago an innovative teacher at the Lincoln School for the educationally and physically handicapped in Boise, Idaho, introduced Leslie to a microcomputer. And suddenly, for the first time since her accident, Leslie could start taking control of her own life.

Jim Schnur, who is a consulting special education teacher in the Boise district, knew Leslie and the computer were made for each other from the start. "We could hardly pry her away from it," he said.

Leslie's life changed dramatically after the horseback riding accident, but it may have changed even more dramatically when she found she could express her innermost feelings to people once again by typing them on the computer.

According to her mother, Sue Evans, the computer has made a world of difference.

"It gives her a really good feeling about herself," says Mrs. Evans about her daughter. "Independence is an important thing."

Mrs. Evans admits that she would sometimes interrupt Leslie while she was using the lap board to spell out a sentence. "Instead of allowing Leslie to finish it, I'd look at the halfway completed sentence and give my version of what I thought she was trying to say."

The computer allows Leslie to initiate conversation and map out a display. It's much easier to operate than a regular typewriter. Another person does not have to be present while she is writing her dialogue.

One of the first messages Leslie typed was to Mrs. Evans and it was blunt: "Mother, you're a motor mouth."

"And I know it's true," Mrs. Evans recalls with a laugh. "The way I spoke for her was probably not the way she would have wanted something said. The computer has given her the possibility of saying things her own way."

Her Apple II has also allowed Leslie to be a more normal daughter to Mrs. Evans and her husband Dean.

"She told her dad off the other day in no uncertain terms," says Mrs. Evans. "I think it's great that she no longer has to keep her emotions bottled up inside of her. This allows a release."

Before the microcomputer, Mrs. Evans found it difficult to evaluate her daughter for mental retardation since she could not communicate verbally. But Leslie's teachers maintain that her mind is normal, and the computer helps demonstrate this.

Schnur reports that Leslie is doing very high level academic work. Her spelling and punctuation are perfect. The only problem with her writing is that it's *too perfect.* Since most people converse with each other in phrases instead of formal sentences, the staff at Lincoln is trying to get Leslie also to use more casual language.

Instead of answering "Yeah, I'm okay" to a question about her health, Leslie will usually write, "Yes. I am doing fine today."

"Sometimes her language on the computer sounds like a business executive," Schnur says.

Besides her adaptability to the computer, Schnur believes Leslie's above-average intelligence is evident in her ability to listen intently and in her understanding of even the subtlest forms of humor between adults. "She doesn't miss anything, especially when it comes to humor," he says.

No one doubts that Leslie has come a long way. She was injured when she was sixteen, the day she returned from a week-long 4-H riding clinic in Alpine, Wyoming. Leslie was riding a new, faster horse when it stumbled in a pasture and sent her flying through the air.

She landed on her head and was in a coma for a month in

Saint Alphonsus Hospital. She spent another month and a half there, two months followed in a nursing home, and later three more months in Elks Rehabilitation Hospital.

Leslie enrolled at Lincoln school in September 1979, where she was placed in a class with much younger students because of the extremity of her disabilities. She was one of the most physically handicapped students Schnur had ever seen, and after a few months, he decided to take a chance.

Schnur thought a microcomputer such as the one he read about in an education digest might help. "She had been here long enough," he remembers. "None of the traditional approaches to teaching worked with her. A computer was our only chance."

The computer provided Leslie with the means to be in a regular classroom with students her own age. "She was an honor student prior to her accident. The lower level classes just weren't stimulating enough for her."

Lincoln School principal Bob McIntyre agreed to the project, federal funds were taken from Title 6B, which provides support for the handicapped, and a $2,500 microcomputer was purchased—a Bell & Howell Apple.

"We were in the process of experimenting with Leslie," McIntyre admits. "We wanted to see if a computer was plausible for others with her type of handicaps."

Schnur says he never looks for miracles with the severely handicapped. "We don't look for progress in leaps and bounds, but if you could have seen what she was like when she came in and how far she has progressed since, it's definitely remarkable."

A total team effort of individuals at Lincoln School and of others in the Boise community helped make Leslie's story so successful.

Her days at school begin when teaching aide Sharon Swarner helps her off the bus. At the beginning of her lessons, Leslie is put up to the computer and asked if she has something to say.

After not being able to speak for two years, Leslie had lost the art of "just talking," said Swarner. But, fortunately, Leslie is an avid letter writer, and this is slowly helping her regain conversational skills.

Almost as if she is making up for lost time, Leslie tries to write letters to everybody who writes to her. She writes thank-you notes to people who have donated money for her home computer and electric wheelchair, and she recently wrote to a former teacher she had during the period she couldn't communicate and told him about her new computer.

Math skills also have to be relearned, although Leslie claims she never knew her multiplication tables very well even before the accident. English, though, is her strong point. Leslie took creative writing courses in high school before her accident and is starting to show interest in this again. "Little Speckled Mule Deer Fawn" was written from the viewpoint of the animal, according to Leslie.

She also continues in her love for books. Mrs. Evans has access to a special tape library for the handicapped. Although Leslie prefers to read light, humorous books, the library is in short supply of these. She has been able to reread some of her old favorites, including *To Kill a Mockingbird, Black Beauty,* and *Alice in Wonderland.*

Phys. Ed. will always be one of Leslie's favorite subjects. She can use the swimming pool safely while wearing a life jacket, and there is also a special bowling alley. To bowl, Leslie approaches a ramp that is parallel to her arm, nudges the ball, and sends it down the alley. Swarner says that recently she and Leslie had the highest team combination bowling scores: 127 and 138.

"I spend all day with Leslie, and I find her amazing," says Swarner. "I just marvel at how well she has adapted. I don't think I could have done it."

Although Swarner reports that her student had temper tantrums when she first started at Lincoln, she says Leslie now smiles all the time. "She has a sunny disposition. She loves to laugh. You don't have to talk to her—you can just look at her and know she's a nice person."

Pam Dunbar is Leslie's occupational therapist at Lincoln. She helped position the computers and found a suitable table and chair so Leslie could run them with the least amount of effort. Leslie has limited movement in her head and trunk, and very little automatic control. She has to remember constantly to do basic things like holding her head up and swallowing. Both teachers agreed the computer has helped with her therapy because she is forced to keep her head up to use it.

"She's very brave," notes Swarner. "Sometimes it's really painful for her to raise her head, so we explain why it's so important. She answers back on the computer that she understands why she has to do it, and then, *she'll do it.* She has extreme patience with us."

Because Leslie has limited motion in her right hand, a special metal plate was put over the computer keyboard to isolate each letter. Sometimes Leslie accidentally drags her arm over the keys, but the plate enables her to hit only the correct letter.

Dunbar also managed to wean Leslie away from a pencil she once used to facilitate her typing. Both Dunbar and Leslie's physical therapist decided Leslie didn't need to use the pencil to hit the keys, and after more involved training, Leslie learned to use her finger to type.

After Leslie became proficient with a computer at school, it was apparent that a similar device was needed at home. But finances were tight for the Evans family; ironically, Leslie's father had brain-stem cancer five years ago that left him also disabled. And that's where Jane Foreman entered the story.

Foreman, marketing director at Northwest Computer Co., had met Lincoln teacher Schnur through the store. "He started telling me about Leslie, and I decided I wanted to do something for her," she recalls.

Foreman got together with the Boise *Idaho Statesman,* and a front-page story ran in the newspaper on Thanksgiving morning. The fundraising began, and Foreman brought KIVI Channel 6 of Boise to film Leslie working on a computer. For the television story, Leslie wrote, "This is my computer. This is how I communicate, and I want one at home."

Contributions started pouring in, and, says Foreman, the press coverage helped to keep the fundraising alive. "A couple of the reporters really became attached to Leslie."

Donations from private individuals and civic groups ranging from a few dollars to a few hundred dollars continued to arrive along with "some really neat, touching letters," Foreman says.

By Christmas there was enough to buy an Apple II computer for Leslie to use at home, and a local Basque Club had also pitched in with a $2000 gift certificate for an electric wheelchair for her. At the time, she was being pushed in a manual wheelchair, and the new chair gave her considerably more freedom.

Mrs. Evans says her daughter was very frustrated before the computer and still experiences some frustration over the length of time it takes her to type out her messages.

"Leslie's gaining a little in speed, but she's still so slow and she tires fast. I think she puts in so much effort.

"She does feel much better about herself, though," said Mrs. Evans. "And if the computer didn't help her with anything else, at least it's doing that. We're fortunate to have it."

Mrs. Evans believes the computer has a lot of potential for her daughter, perhaps even as a means of employment someday. She knows this isn't feasible right now because Leslie works so slowly, "but down the road someday, who knows? They're making so many advances with computers."

At home, Leslie uses the computer for basic communication. She tells her mother what she did that day at school, and then does her schoolwork, writes more letters, and plays video games. She had a duck hunt game, but became bored with that. Leslie would rather play card games, and her mother warns that she is fiendish at blackjack—most of the time.

"She was playing with some friends the other night and thought she was ready to go to Las Vegas. But when I looked in on the game twenty minutes later she was losing quite heavily. Good thing she didn't go."

Leslie has recently learned to eat by herself, and although Mrs. Evans reports this is a messy process, for Leslie it's one more step toward independence.

"We go out for pizza and she doesn't mind if people stare. She doesn't seem to relate to being handicapped," says Mrs. Evans.

Besides eating, other everyday functions, such as dressing, present big obstacles for Leslie. It takes her 45 minutes to an hour to take off a coat or jacket. Now she is also capable of unzipping her boots.

These basic functions and homework take up most of her evening, so Leslie and her mother are looking forward to summer when they'll both have more spare time for other activities. The Evans family recently acquired a van to take Leslie on outings.

"Not long ago I asked Leslie what she would do if she could do anything," Mrs. Evans says. "She told me she wanted to go for a ride."

Mrs. Evans thought Leslie was talking about the van, but her daughter replied, "No. On my horse," and then she started to cry.

"That's one of the few times I've seen the accident affect her," says Mrs. Evans.

But Leslie continues to make progress. One of hardest things for Mrs. Evans to learn was that her daughter was not going to allow her to become an overprotective parent. When she left for cerebral palsy adult camp last summer, Leslie went out the front door and didn't even look back at her tearful mother.

Schoolmates of Leslie also take her to the movies, and Mrs. Evans has learned not to worry about her daughter.

Because of the success with Leslie and the computer, Lincoln principal McIntyre would like to have more computers for use by other students. The school is planning to purchase two more microcomputers by next fall.

"Our aim is for computer literacy. We want to expose the students to a society that will even be more computer-based in the future," he says.

McIntyre explains that computers help the students with their drill and repetition work. Although, he emphasizes, computers could never replace a teacher—they do enhance instruction. For example, if a student is reading at a fourth-grade level and that proves to be too complex, a specially programmed computer can automatically drop the same material to a third-grade reading level.

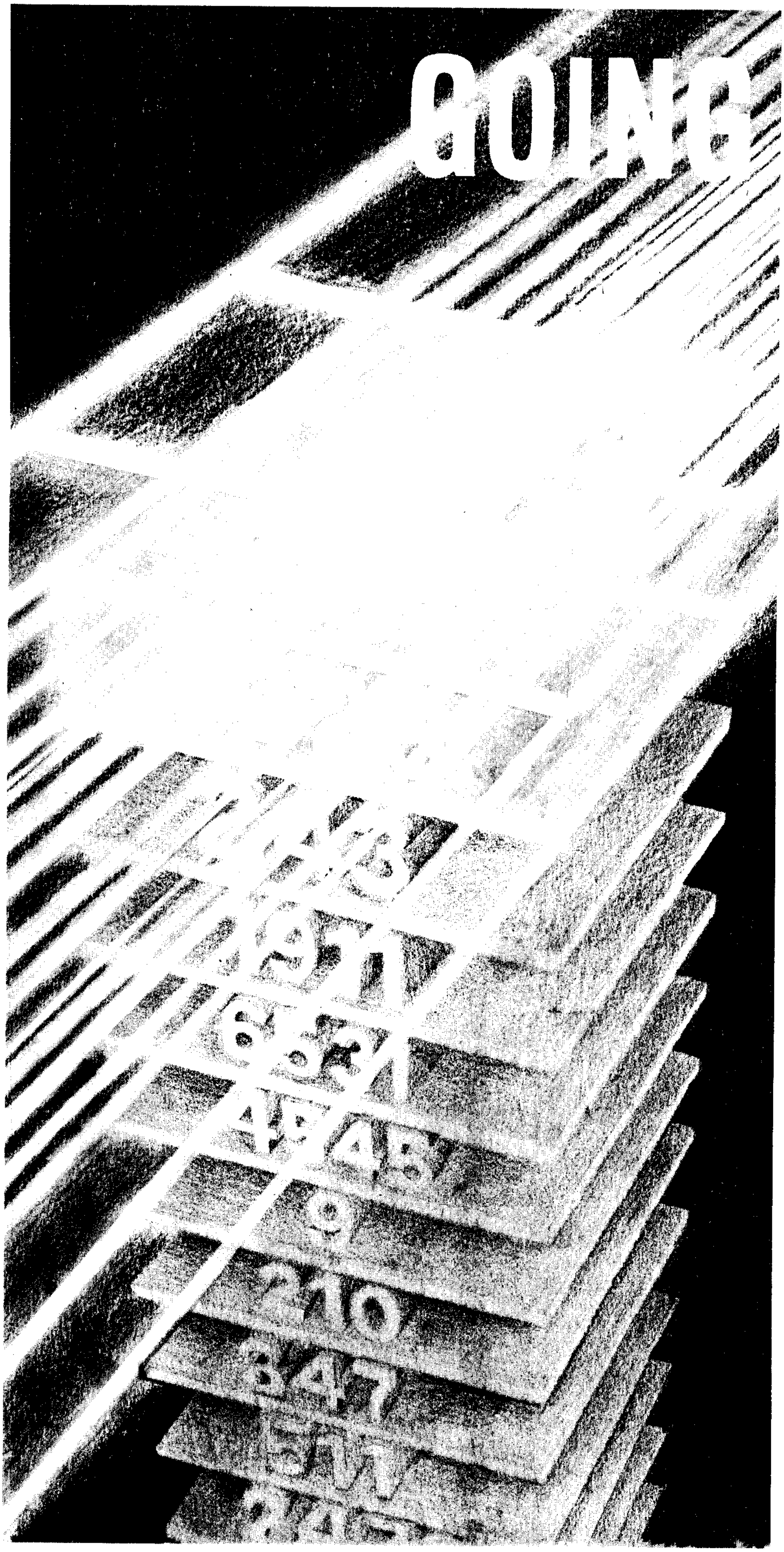
The students at Lincoln range in age from six to twenty-one years, and they are provided with a full day of special education. Vocational programs are also offered, including training in automobile maintenance, industrial laundry, and fast food service. Lincoln has its own restaurant where the students learn to wait on tables, cashier, and cook. A computer will eventually be used in the food service program for inventory and cash register sales.

Like Leslie, the other kids at Lincoln are "totally turned on by the computers," says Schnur. "It's a far easier means of written communication, especially for the verbally handicapped."

"If you had all of this inside of you like Leslie, the creative writing, the letters, the emotions—and no way to get it out, can you imagine where you'd be?"

Maybe Leslie said it best when she typed for *Softalk*:

"The computer makes it easier for me to express my feelings and thoughts, to keep in touch with friends and family. I use it for homework, games, etc. It cuts out a lot of frustration on both Mom's and my part, and it is fun. Leslie Evans. May 4, 1981."



GOING FORTH

BY CRAIG STINSON

Forth. The name itself suggests movement and immediacy, directness as well as direction—as in going forth, forthrightly, forthwith.

The connotation is probably accidental. Charles H. Moore, Forth's inventor, first named this unique programming language in an implementation for an IBM 1130. He wanted the name to stand for "fourth generation programming language," but since the IBM only allowed five-character identifiers, "fourth" got contracted to "forth."

Nevertheless, immediacy, speed, and straightforwardness are salient qualities of this language, helping to account for its appeal to a growing number of programmers.

The Eye of the Needle. Forth is usually described as a threaded language. At its core are some fifty operators coded in the machine language of the host processor. The rest of system consists of an additional hundred or more operators (called words) that are defined in terms of the original fifty.

Programming is a matter of extending Forth's vocabulary. Simpler procedures are used to define more complex ones, and these in turn can be tested, debugged, and regrouped into still larger units, until the entire program becomes a single word.

If this so far sounds Pascalish, hold on. Forth has some dramatic differences.

First, every new procedure in Forth acquires a status coequal to everything already in the language, so Forth is endlessly extensible. When you define new words, they become a permanent part of the system, unless you choose otherwise. All of your previous work is immediately available to you if you want it.

The language is not only extensible but contractible as well. A word called *forget* will tell the system to drop an item from its vocabulary. Forth is compact to begin with—the language usually fits into 16K or less—but you could easily reduce its size, streamlining it for a particular application where memory is at a premium.

2 + 2 = Dandelions, If You Like. Besides being able to add to or subtract from Forth's vocabulary, you can redefine it. As trivial examples of this flexibility, two could be renamed three, and plus could become minus. The point is that the language is malleable and can be customized to suit the purpose at hand.

Another major difference between Forth and many other high-level languages is the absence in Forth of error-checking routines. Nothing inherent in Forth warns the programmer against mixing data types or writing array subscripts out of bounds. Of course, if you wanted such error protection, you could program it in. In Forth, naturally.

To a person looking at Forth code for the first time, its most striking features are likely to be the use of postfix notation and the way it manipulates a data structure called a stack.

Postfix notation, affectionately known as reverse Polish notation, is the style of input used by Hewlett Packard calculators. In RPN the operands are entered before the operators. Instead of a formulation like 5 + 9, in Forth you would input 5 9 + . To print that sum you would put a period after the plus sign; the period is an operator signifying "print."

For a simple expression like 5 + 9, this may seem gratuitously weird, since it's so contrary to the syntax of our natural language and thought. RPN begins to look sensible when you get to more complex formulations like "print (8 + 17) * (42 + 9)." That would be reverse Polishly notated as 8 17 + 42 9 + * followed by a period for print (omitted here for typographic clarity). In RPN, you never need parentheses.

A Stacked Language. To understand reverse Polish notation, it's necessary first to understand the concept of a stack.

A stack is like a queue, except that every newcomer goes to the head of the line instead of the back. The stacking mechanism is commonly compared to that device used in cafeterias to pile a column of plates on a spring; anytime someone takes a plate off the top, the rest of the column gets bumped up a notch.*

The stack is a last-in, first-out (LIFO) structure; data can enter or leave it only at the top. The operations of putting data on and removing it from the stack are known as *pushing* and *popping*, respectively.

To go back to our first example, here's how RPN works: when you type 5 9 + , first the five gets pushed onto the stack, then the nine (above the five), then the plus operator pops both numbers, performs the addition, and pushes the sum back on. Suppose the top three numbers in the stack before this operation were: 42 6 11 (reading downward). After you typed 5, the stack would read 5 42 6 11; after you typed 9 it would be 9 5 42 6 11, and after the plus it would be 14 42 6 11. If you followed the plus with a period to print the sum, the top three levels of the stack would once again be 42 6 11, since the period operator pops the top number and prints it.

Pop the Top. Now follow the more complex example, (8 + 17) * (42 + 9). Reverse Polish notation has it 8 17 + 42 9 + *.

After the first plus sign the top number in the stack is 25; 42 and 9 then get pushed, the second plus sign pops them off, 51 gets pushed, and the asterisk pops the top two numbers, now 51 and 25, and pushes the product, 1275.

The stack may hold data other than arithmetic operands. For example, when you create a string, the string gets stored in memory, and the stack holds its starting address and length.

What's unusual about Forth is not the stack itself—it's a structure found in all programming systems—but the fact that the programmer manipulates it directly. This direct manipulation largely accounts for Forth's compactness. Pascal requires so much more memory than Forth not only because of its error-checking measures but also because it has to parse its source files into—would you believe—reverse Polish notation.

In terms of execution speed, Forth is comparable to Pascal or Fortran. All run much faster than Basic and much slower than machine language. So why should a programmer forego error checking and other built-in amenities if no great gain in speed results?

They Like the Freedom. Part of the reason for using Forth may be an issue of style. Like machine-language programmers, Forth adherents seem to enjoy the relative absence of syntactic constraints and the bare-handed contact with the medium.

As for error checking, the characteristic Forth process of building larger and larger subroutines out of self-contained smaller ones allows the programmer to debug along the way. Forth definitions are compiled as they are entered and are immediately ready to run.

A finished Forth program may be harder for someone else to read and amend than a comparable program in Fortran or Pascal (unless the Forth program is unusually well furnished with comments), but the style of the language encourages the Forth programmer to get it right the first time.

Enthusiasts also say it's easier to optimize code in Forth than in other high-level languages. Areas of a program that are not as efficiently written as they might be are easier to find and correct; and procedures that require machine-language speed can easily be coded in assembler and woven into the Forth tapestry.

Esperanto of Computerese. Another big plus is portability; only a very elemental part of the language is machine specific. Because of this, a software vendor using Forth can make products available to anyone with a minimum of recoding.

The malleability and extensibility of the language itself offer further advantages, particularly in dedicated industrial applications such as controlling machinery.

Forth got its start in such settings. The first major use of the language was for a data-acquisition program on a radio telescope at the National Radio Astronomy Observatory in 1971. Two of the earliest Forth programmers, after Charles H. Moore and Elizabeth Rather, were astronomers at Kitt Peak; in fact astronomers provided most of the early impetus to the language.

Moore has written that he never set out to create a language as such; as a programmer, he was trying to find ways to write more and better programs. Through Moore's work on a number of different mainframes in different industrial settings, Forth gradually evolved.

*If you feel like you're experiencing deja-vu, you really aren't. Coincidentally, Roger Wagner uses precisely the same simile for stacks in this month's Assembly Lines. The near-exact repetition was allowed to stand to illustrate the common base of principle between languages.—MCT

All of its essential features, except for the compiler, were completed by 1968.

Going Forth To Speak to the Stars. Forth Incorporated was founded by Moore in 1973 to exploit the interest of astronomers. Serving astronomers, Moore has said, would still be the company's main line of business if it weren't for the fact that there are so few new telescopes.

Currently, Forth Incorporated provides versions of Forth for a variety of microprocessors (not including the 6502), as well as for various minicomputers and mainframes. The company also supplies extensive support, custom packages, and literature about Forth.

Organized in 1978 by five programmers, the Forth Interest Group, aka Fig, is a nonprofit organization the purposes of which are the furtherance and support of interest in Forth. In the three years of its existence, Fig's membership has grown from five to twenty-five hundred. Roughly 90 percent of the membership live in the United States, and nearly half of the American contingent is in California.

In addition to publicizing Forth at trade shows and through their own journal, Fig recently installed a Conference Tree bulletin board service at (415) 538-3580. Interested parties can dial up and ask questions or exchange comments about Forth, Forth applications, vendors, and so on. Plans are afoot to include abstracts of technical papers on the Conference Tree, so that readers can sample what interests them and buy directly from the authors.

The Conference Tree software was written, in Forth, of course, by John James.

The Do-It-Yourself Version. Fig makes available for ten dollars a public domain version of Forth (called Fig-Forth). It comes as an assembly listing only, and is entirely unsupported. But there are no legal constraints on what you do with it.

Versions of Fig-Forth are available for a number of different computers, including nine microprocessors. There are two for the 6502, one produced on the Rockwell R65 and the other on the Apple using the Ted Assembler. For information write Forth Interest Group, Box 1105, San Carlos, CA 94070.

An enhanced version of Fig-Forth for the Apple II is available for $140 from Cap'n Software. This package, by John Draper, includes all the public domain Fig-Forth plus disk drivers, high-resolution and low-resolution graphics, a screen editor, and manuals. It also includes a macroassembler by William Ragsdale. Cap'n Software's Fig-Forth has been on the market for about a year and a half.

An effort has been underway since the late seventies to produce a standard Forth. A Forth Standards Team, currently chaired by Ragsdale, has held five voting meetings, all in out-of-the-way places like Catalina Island and Utrecht, Holland, so that only those genuinely interested would show up.

Out of these meetings have arisen three standard Forths—Forth-77, Forth-78, and Forth-79. The changes that went into the latest of these, adopted in October 1980, were made with the overriding goal of achieving complete portability for the language.

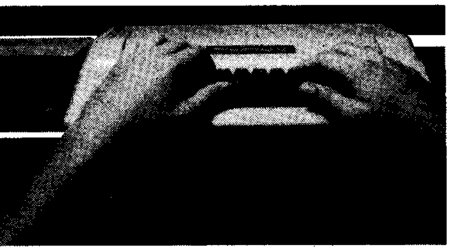
A Third and, Yup, a Fourth Forth. MicroMotion has recently issued Forth-79, coded by Martin Tracy for the Apple. Tracy's Forth comes with a macroassembler, lo-res graphics, and a full screen editor. It retails for about $90.

MicroMotion also sells separately a thorough manual. Ten chapters provide a tutorial in both Fig-Forth and Forth-79. Apple-specific material is covered in appendices.

Softape also offers a version of Forth for the Apple, called Forth II, for about $70. Written by William G. Graves prior to both Fig-Forth and Forth-79, it has a few variations in vocabulary from these later models. It comes with a readable, informative manual, and three demonstration programs (two games and a music sample). Forth II includes a line-oriented editor and vocabulary for lo-res graphics, but lacks a built-in assembler.

A still earlier Forth for the 6502, from Programma, has been discontinued. ■

MARKETALK News



□ If you're not quite ready for another language, but you envy the extra capacity of people with the Language System; if you won't buy DOS 3.3 because you're planning to buy a Language System as soon as you can afford it; if you have a 3.3 *VisiCalc* and no extra memory to take advantage of it—**Apple** has the answer. At last, you can buy the *Language Card*—the hardware part of the Language System—without buying Pascal. Although the card itself won't give you Pascal, it does give you the capacity for it and for Fortran. It gives you 3.3 (if you don't already have it), and 16K extra RAM, which, among other things, gives you the capacity to load and run programs designed for a 64K computer or 56K available program space. It also saves you $300 against the cost of a full Language System. 48K. $195.

□ **Advanced Business Technology** (Saratoga, CA) now offers two programmer's aids for Apple Pascal versions 1.0 and 1.1. Both *ABTools I and II* have file edit/dump utilities, p-code assembler, file comparisons, a cross-reference generator, and binary patch/edit function that operates on disk. Help menu available with each input. *ABTool I* features abilities to compare two text files for length and reediting, paginate, break up large text files for editing, and combine small files together. *ABTool II* can edit arbitrary binary files in hexadecimal, report on any variance in byte position between two files, and generate arbitrary text files including nonprinting characters—permitting, for example, underscoring. 48K, DOS 3.3. $75 each. You can now enter programmable character strings and special characters not provided on Apple's standard keyboard with ABT's *Softkey* keypad. Software drivers provided on sixteen sector disk in Applesoft or Pascal. The fifteen-key pad, the size of a small calculator, plugs into game port. $150.

□ The *Lynx* from **Emtrol Systems** (Lancaster, PA) is a direct modem. Features include number of stop bits, parity, programmable word length, originate/answer, and full/half duplex; autodial and autoanswer features are optional at extra cost. Instruction manual lists bulletin board, Source, and CompuServe numbers. $289.95.

□ The *COMMLOGG* dialer for Hayes Micromodem II from **Harvey's Space Ship Repair** (Las Cruces, NM) has a built-in software clock, keeps track of running time of each call, calculates charges, logs information with the day's date (which user inputs daily with built-in calendar routine), and provides cumulative monthly charges. Autodial function allows fast dialing of often-used numbers. Dialer allows use of standard or military time, calculation of holiday rates, and printing of autodial and billing files that can be edited with built-in text editor. 48K, DOS 3.3, ROM Applesoft. $39.95.

□ *Letter Perfect* from **LJK Enterprises** (Saint Louis, MO) is a word processor that doubles as a data base. Permits data base merges, headers, footers, search and replace, tabbing, scrolling and scrolling speed changes, moving of text blocks, boldfacing, and right-hand justification. User has wide berth for formatting text; screen format feature lets user see text onscreen as it will appear on paper. Cursor for editing moves a page at a time—backward or forward—or to beginning or end of line. Compatible with most eighty-column boards; in revised version, that includes the Australian Vision-80. 32K, DOS 3.3. $150.

□ A different style in diskette cases comes from **Southwestern Data Systems** (Santee, CA). Velcro flaps open out, turning case into stand-up flexible file with plastic envelopes. Flaps fold up, lid snaps shut, and you have a neat, compact carrying case. Case holds twenty diskettes; comes in sizes for 5¼-inch or 8-inch floppies, $19.95 and $26.95, respectively.

□ A type ahead buffer, the *Model 150* from **Vista Computer Company** (Santa Ana, CA), has a type ahead capability of as many as forty characters. Buffer lets you type in commands or data while Apple is processing previous information. Requires no software patches, cuts, or jumpers. Manual guides you through easy installation. $49.95.

□ Software that plays games and speaks in tongues is published by **Synergistic Software** (Bellevue, WA). *Tank Attack* pits you against another player in command of combat tanks in your choice of five scenarios varying in difficulty. Beginners battle like Rommel and Patton in the desert; with warfare experience, you progress to the mountains, forests, and cities. In *Death Run*, the computer's gremlins have invaded Earth, and you and your pursuing scout car must track them down and wipe them out before they conquer the planet. Both games on one disk. $20. *The Linguist* is a translation and tutorial package that teaches pronunciation, spelling, and definition of words or phrases in English, Russian, Japanese (Katakana), Hebrew, Greek, or other languages employing the same alphabets. International phonetic alphabet and other phonetic symbols, hi-res graphics for printout, facility for constructing customized word lists for drilling, special Romance language characters are among program's features. *Linguist* has storage capacity of 4,400 words, 2,600 definitions, and 2,000 foreign phrases. Allows backup disks. $40. Both products require 48K, DOS 3.3, Applesoft.

□ The *Job Cost Accounting System* from **Charles Mann and Associates** (Yucca Valley, CA) lets you create as many as a thousand general ledger accounts and ninety-nine job cost accounts, with payables automatically posted to either account. Also provided are text processor and a report generator. Payroll portion of program constructs weekly, biweekly, semimonthly, and monthly payrolls; calculates tax information; and prints checks, monthly deposit reports, audit and filing reports. 130-column printer, two disk drives, 48K, DOS 3.3. Introductory price, $359.95. After June 1, $459.95. Available for Apple III.

□ The *APMOD* board from **Connecticut microComputer** (Brookfield, CT) generates a bus from the Apple, allowing computer to read voltages, light levels, temperatures, and pressures and to control home appliances, heaters, lamps, pumps, and motors. Board has decoding logic, eight-bit tri-state buffer, and eight-bit latch. $59.95.

□ **WIDL Video** (Chicago, IL) releases a summer/fall edition of their software directory for the Apple. Directory is divided into three volumes: first volume lists business software, including all new word processors and data base packages; second volume is devoted to game software; volume three guides reader through fast growing educational market and includes a special section on Apple utility packages. $5.95 per volume.

□ **Westware Software** (Ontario, OR) has made their *Systems II* business software usable with the Corvus 3.3, ten-megabyte disk operating system, as well as with Apple's DOS. Multidisk package updates and sorts accounts payable and receivable and inventory, maintains general ledger and payroll files. Included is a KSAM card that eliminates sorts after entry and allows high-speed search on the key field. Requires 80-column board, 132-column printer, 48K, $1,295.

□ Three packages under the title *Pro-Soft Billing System* from **Professional Medical Software** (La Crescenta, CA) help manage medical accounts receivable. Features include accounting for as many as seven thousand active patients; printouts of MediCal, MediCare, and Universal AMA insurance forms; billing from up to ten doctors; mailing labels and patient listings sorted alphabetically or by account number; and accounts receivable aging and delinquency reports. Version 1.0 has private party billing only; version 2.0 has private party billing with insurance billing. $995 and $1,995, respectively. Version 3.0 has insurance billing and runs on Corvus hard disk. Eighty-column printer, 48K, DOS 3.3. $1,995.

□ The *CTA ADC-16C* board from **Computer Technology Associates** (El Paso, TX) converts the Apple into a digital voltmeter that can measure temperature, light intensity, water pressure, among other variables. This sixteen-channel, eight-bit board converts analog to digital information at a conversion time of one hundred microseconds per channel. Output is latched tri-state; input is latched address. Test connector, software, manual included. $199.

□ **Shafer Software** (Sunnyvale, CA) offers educational software for the youngest of students with *Pre-Schooler*. Package teaches alphabet and how to distinguish between letters in two ways: in list form, and in a Concentration-type game in which student wins by remembering letters and beating a tree-eating monster. Only hi-res big letters are used. 48K, DOS 3.3. $30.

□ An assembly language development system comes from **Apparat** (Denver, CO). The *APEX* system has an editor and an assembler. TEC65 editing language corrects and creates ASCII text files and can be used to edit manuscripts, letters, programs, or hexadecimal memory dumps. It does macro commands, string search, cross buffer search, and special inserts, and has ten text buffers and command registers. Assembler is a two-pass resident assembler for 6502 machine code. It provides twelve pseudo-ops and twenty-three sixteen-bit expansion op-codes, as well as offset assembly, forward references, and 1,900 lines per minute speed. 48K, DOS 3.3; also available on eight-inch disk. $99.

□ A fan to cool down the Apple during the summer heat is manufactured by **M.R. Engineering** (Chicago, IL). It mounts on power supply; air vents are positioned flush to Apple's side slots. Silent running fan available in either 115-volt type or 220-volt model. $45.

□ The applications of microcomputer-controlled robots is the theme of a bimonthly magazine, *Robotics Age*. Robots of every class, from heavy laborers to nimble thinkers, are the main characters of each issue. Topics covered include robot circuit design and software, industrial robots, CPU architecture for robot control, and methods of robot reasoning. Year subscription, $15; two years, $28. Write Box 801, La Canada, CA 91011.

□ *Blocks Author*, a CAI authoring system for teachers, has been developed by the **San Juan School District** (Carmichael, CA) and the **California School for the Deaf**. System consists of twenty-five disks: twenty comprise a graphics library, five are authoring programs that use the library. Concept is to allow teacher to develop curriculum and programs coupled with graphics. Included is a data management program that acts as a lesson planner and monitor of student progress. Two disk drives, 48K, DOS 3.2 and 3.3, Integer. $500.

□ A word processor with single key commands from **Select Information Systems** (Kentfield, CA), *Select* boasts easy text manipulation, extensive help menu supplemented with a computer assisted tutorial disk that replaces hardcopy documentation. Special feature is *Superspell*, a 10,000-word dictionary that proofreads text and displays typing errors and corrections. Requires SoftCard, two disk drives, 48K RAM, 3.3. $399.

□ The *CORP Data Base Management System* from **Maromaty and Scotto Software Corp.** (Bellerose, NY) writes data base systems in Applesoft Basic. *CORP* generates programs that can be saved onto a separate disk and can be executed independently of the *CORP* master disk. Package includes sort and update subsystems, a print applications generator, data entry applications generator, and diagnostic package. Apple II or III, two disk drives, 48K, DOS 3.3, ROM Applesoft. $189.95.

□ **Creative Computing** (Morristown, NJ) staged a national software competition; winning programs would be published and marketed by the company. Two came out on top: Ken Murray's *Streets of the City* and Richard Galbraith's *Trucker*. *Streets of the City* puts you in charge of a ten-year plan to improve streets and transit service; at the same time, you must stay in the good graces of a majority of the city commission that appointed you. You haul cargoes coast to coast in *Trucker*. Balance risks and benefits with type and weight of load transported, weather conditions, amount of sleep, driving speed. Above all, don't get behind schedule. 48K. $24.95.

□ *Visi-Caids*, a *VisiCalc* utility from **Data Security Concepts** (Manchester, MO), alters text files according to your specifications. Program is divided into three sections: Label Splitter divides wide alphanumeric columns into two or more shorter columns, improving readability of printed reports; Formula Reader displays or prints formulas, numbers, and labels in rows or columns, including coordinates; PD Reader prints or displays *VisiCalc* "/PD" test files on 16K to 48K Applesoft machines without booting *VisiCalc*. DOS 3.3, $34.95.

□ The simplified version of *The Complete Graphics System* is *Magic Paintbrush 4.0* from **Co-op Software** (West Chicago, IL). Game paddles control drawing module that you use with lines or digital paintbrushes. New version has improved shape table designer that lets you make Apple shape table with keyboard or paddles. Five games created with *Magic Paintbrush* are included: *Color Invaders, Slot Machine, Sailboat Race, Dogfight*, and *Collision*. 32K, DOS 3.2 or 3.3, Applesoft. $29.95.

□ Hand-eye coordination and visual judgment are tested in *Perception 3.0* from **Edu-Ware Services** (Canoga Park, CA). User can adjust difficulty level on seven separate programs. Performance is reported by a scoreboard, which also maintains a cumulative progress report. 48K, DOS 3.2, ROM Applesoft. $24.95. The latest member of Edu-Ware's stable of math programs is *Statistics 3.0.* Program allows quantification and evaluation of mathematical relationships. It features edit,

GOTO 78

MARKETALK

Reviews

Ultima. By Lord British. Once in a while, a product appears on the market that truly seems to reach the ultimate development of its genre. The phenomenon occurs more often than usual in this infant industry of microcomputing, and, also peculiar to this industry, such products are often surpassed in relatively short order.

Aptly named *Ultima* fits the first description but is likely not to succumb to the second for some time. There is no doubt that eventually it will be surpassed, however, if only by its own author: Lord British, whose identity is detailed elsewhere in this issue, is merely a sophomore in college. What we can expect from him in years to come is beyond imagination.

Ultima is a role-playing adventure game of the hi-res, set-command variety rather than the text, deduce-the-vocabulary type. Although there are dungeons in *Ultima*, they are only a portion of the program. Through the rest of the game, your character can roam an entire aboveground world. Only Bob Clardy, with his *Wilderness Campaign* and *Odyssey*, has paved the way for this style.

There are continents, lakes, rivers, oceans; there are towns and castles and, of course, dungeons. Every castle has an individual ruler to serve by fulfilling quests, an imprisoned maiden you may be able to rescue, and a jester who may steal all your possessions. Each castle also has plenty of guards.

Each town has six enterprises: shops for armor, weapons, means of transportation, food, and magic; and a bar. As in real bars, the bartender can be a source of important information; but the bar wench, out to seduce you, will roll you for all your money if she succeeds. The town, too, has guards, although not nearly as menacing as those in the castles.

Rulers, which include one Lord British, give you rewards for your questing; the nature of the rewards range from advice to extra strength. You must finish one quest for each ruler to win.

At the beginning of the game, available weapons are swords and daggers; armor is leather, chain, or plate; and forms of transportation are horse, horsecart, raft, and frigate. Later, other items gradually become available—which leads to the ultimate uniqueness of *Ultima*: although the game begins in medieval times, as you grow in experience and strength, the times progress, technological development makes new products available, and, eventually, you reach the space age.

Even then, *Ultima* is not through, but what lies beyond the space age is part of what you must discover through playing the game. It lives up to what precedes it.

The dungeons look almost like those in the Lord's earlier work, *Akalabeth*; a few of the monsters are similar as well. There are plenty of new, imaginatively conceived monsters; and subtle changes in the construction of the dungeons make them clearly second generation. For example, from the side, the ladders look like sideways ladders, and you can't climb them until you move to the front or back. There are no seams in the walls, and you can detect secret doors ahead of time and make them permanent if you like. Ladder-up and ladder-down spells are buyable and relatively inexpensive, and their efficiency depends on your characteristics.

All your character's traits can be bettered, although its type—fighter, wizard, cleric, and so on, along with its inherent characteristics such as the ability to cast certain spells—cannot.

Ultima comes with two disks, a master game disk and a master player disk. The game disk is protected and you'll use it for all play. The master player disk is intended to be used only to make copies of it. Each copy will contain one character you build and one game you can save over and over as you go. Don't play on your master player disk, or you'll be forever confined to your first character.

Ideally, you'll play *Ultima* with two disk drives, but it isn't necessary. It merely saves occasional disk-changing.

In *Ultima*, Lord British has created a fast-paced, detailed, imaginative role-playing masterpiece in full-color hi-res. If you like this kind of game at all, this one isn't to be missed. And if you like space travel and skill games—you'll get that here, too, although you'll have to work for it.

This review does not begin to cover all the events available in *Ultima*, and it gives you no strategy. A lot of the fun is discovery, and there's much to discover. MCT

Ultima by Lord British, California Pacific Computer Company, Davis, CA. 48K, DOS 3.3, one or two drives. $39.95.

Vision 80. By Ken Thompson and Harry Harper. When Harry Harper blew into California from Sydney for the West Coast Computer Faire with a new eighty-column board up his sleeve, he hadn't even seen his competition. When he did, he knew he had a winner.

He was right. Distributor Vista Computer Products in Irvine leaped at the rights and immediately changed the board's name from the *ZEV* from Zofarry to *Vision 80.* They gave up a colorful name for dullsville, but that couldn't affect the quality of the product.

Harper and partner Ken Thompson began planning their board with one premise: it must be to Apple standards; it must be totally compatible with the Apple and its commands. To their thinking, their board was unfinished until it met this condition. Now it does.

Vision 80 is a long, blue board that is totally invisible to your Apple until you want it. Then you merely call it as you would your printer or disk drive: PR#3. It is slot dependent; that's the trade-off for having it so friendly. When you call, it appears. You may then do anything you like just as you would normally, but you'll be doing it in eighty columns. You may program using VTABs; HTABs; GOTO x,y; NORMAL; INVERSE; HOME; and all other commands. In fact, you may use HOME anytime, and it will do exactly what it's supposed to do. FLASH, however, gives inverse without a flash.

When you want to look back at your forty-column screen for some reason, Control-T takes you there instantly and toggles you back to eighty when you're ready. Editing using ESCape and the A, B, C, D and I, J, K, M diamonds works normally. The regular pokes will change your screen window to whatever dimensions you like within the board screen.

Assembly language programming is not left out. Documentation gives the exact locations used by the card and the single card initialization code. With slight modification, Programma's LISA assembler can be used with Vision 80.

Now that you begin to get the picture of a hassle-free eighty-column board, consider its additional features. Vision 80 gives you upper and lower case automatically using the shift key for capitals and the shift and control keys for shift lock in either upper or lower case. A bell signals that you've changed locked case. When you're in caps, several special characters are available to you, including underscore, various brackets, square root sign, and even a tiny, if skinny, bitten apple.

Lower case characters have strong, clear descenders. The entire character font, squished as all eighty-column fonts are to some degree, is easy to read, although, if you're going to do most of your computing in eighty-column mode, you'd be wise to adjust the width setting on your monitor or TV to extend sideways a tad.

Vision 80 has an alternate character set or a graphics character set built in. One or the other can cohabit the board with the regular character set. With either of these sets, you can toggle back and forth between it and the regular font with a single control key. You can build your own character sets with the graphics set.

When you're programming, your listings will still appear in forty-column format — unless you toggle Control-L. Then even programs you wrote in forty columns will list in eighty.

There's a limit to everything, and Vision 80 has a time when it quits being invisible. If you're running Pascal, Fortran, or CPM, your system will boot in eighty columns, upper and lower case. There is one minor problem here. Vision 80 always boots with Pascal, and if you boot a runtime Pascal program intended for forty columns and not expecting the board, you can get a blank eighty-column screen; unless the program gives you access to ordinary computer commands (rare), you won't be able to toggle back to forty. The solution here is to use your old monitor hookup instead of the hookup that goes direct from the Vision 80. Then everything's fine. Actually, you can use two monitors at once with the board, having the eighty-column screen on one and the forty-column on the other.

Vision 80 automatically switches to the normal Apple screen for graphics, either hi-res or lo-res, and switches back again for text. You can put graphics up on one monitor and run eighty-column text on another. This is especially useful when programming Pascal Turtlegraphics.

There are still bonuses. Vision 80 has a built-in communications driver compatible with acoustic couplers or modems with the Apple Communications Card, the California Computer Systems asynchronous serial card, or any serial interface with the same protocols as the CCS card. Floppy disks or the Corvus hard disk can be used. Vision 80 allows up to 1,800 baud without any data loss. Communications can be Apple to Apple direct or via phone link or Apple to most mainframes. You reach the communications facility with—what else—a single control key.

If your communications setup is not compatible with this, Vision 80 will merely disappear and not interfere with your normal communications in forty-column mode.

Most publishers with software intended to run with eighty-column boards have not provided for the Vision 80; but many, such as LJK Enterprises (Saint Louis, MO), which puts out the *Letter Perfect* word processor, are quickly making provision for it now. Word processors that already provide for Vision 80 are the Computer Solutions *Word Master* and Sandy's *Wordprocessor.*

All in all, it's good to find an eighty-column board that says, at last, you *can* go HOME again. MCT

Vision 80 by Ken Thompson and Harry Harper, Zofarry Enterprises PTY. LTD., Haberfield, N.S.W., Australia; distributed by Vista Computer, Irvine, CA. Communications mode requires 48K. $350. Alternate or graphics character set is included.

Pool 1.5. by Don Hoffman, Howard de St. Germain, and Dave Morock. *Mississippi Belle unconsciously polished her long deep-red nails on her silk lapel. She looked unconcerned, even a trifle bored, but this was artifice. In truth, every nerve in her body was taut with anticipation as she watched the champion line up his shot.*

Dakota Pudgy concentrated unwaveringly on the line from cue ball to eight ball to pocket: a relatively simple shot, except that the world championship hinged on it. Turning away from the table, he relaxed his eyes, breathing deeply, calming himself, preparing himself.

Finally, Pudgy turned back to his task. Belle started at the motion, slightly betraying her cool. Pudgy glanced at her only for an instant, but the break didn't escape him. His confidence doubled. He positioned his shot, chose his timing, and, with total aplomb and bottom English, hit the paddle button.

The cue ball hit the cushion, caromed obliquely toward the eight ball, hit it, and stopped. The eight ball rolled slowly to the edge of the pocket, hesitated, and fell in.

Belle fainted.

Dakota Pudgy smiled his slow, broad smile. Turning toward the pool table on the monitor, the still-reigning champion said, "You didn't really think I might lose, did you, Mama?"

Then he packed up his Apple and, with his arm firmly around it, left Belle, the pool hall, and the town forever.

Don't blame *Softalk* for this fable. It's just the effect IDSI's *Pool 1.5* has on people. It's so realistic, you begin filling in the details of the pool hall yourself.

Pool may mean trouble in River City, but on the Apple, it's pure fun. One of the better physical simulations (nothing yet tops *Apple Bowl*), *Pool 1.5* presents a pool table top, a set of balls, and an extended cue ball — the extension of a dummy transparent ball taking the place of a cue—that work close to just like the real thing. With capacities for general aiming, fine aiming, speed control, and all types of English, you can make most of the shots you could on a real table.

Exceptions occur. On shots where the object ball is far away from the cue ball, the fine tuning isn't always fine enough; you can't draw the cue ball the length of the table with English; and you can't shoot around a blocking ball with English. Most of us couldn't manage the last two anyway. The rest of the time, it's right on.

There are advantages to computer pool. You can get instant replay of your shot, in slow motion if you like, thus learning from a chance play how it might have been executed on purpose, for example. You can control your surface, choosing heavy felt with a high amount of friction, where the balls practically stop as soon as they're hit; bare table, with almost no friction, on which the balls seem to keep mixing and going forever; or any of three in-between settings. You control the speed of play, without affecting the physics of the shots, to be extremely fast—fun for creating executive bouncing balls, aside from pool—or extremely slow, as for a slow-mo instant replay, or in between.

Four games are offered on the colorful hi-res pool table: straight pool, eight ball, rotation, and nine ball. In the last two, the balls appear in white with numbers (except the eight ball, which is white on black in all games); in the first two, they appear in two colors representing the high balls and the lowballs. In any game but rotation, you can toggle to the alternate view of the balls. The cue ball is always solid white.

At any time, you can hit escape to see the scoreboard. As long as you stay with one game, the scoreboard records balls pocketed by each player in the current game, total balls pocketed per player, and games per player. It also notes who is after which balls in eight ball.

Next time you're too lazy to meander to the local pool hall for a beer and a little shooting, go to your refrigerator and turn on your Apple instead. MCT

Pool 1.5 by Don Hoffman, Howard de St. Germain, and Dave Morock, Innovative Design Software, Las Cruces, NM. 48K, disk, paddles. $34.95.

Echo II Voice Synthesizer. By Milo Street. Although this isn't its primary purpose by any means, the Echo II is one of the most fun products to arrive on the Apple scene yet. It is the first voice for the Apple that is really generated by the Apple. In all the others, people speak words into a recorder, and the computer stores them in enormous amounts of memory to be re-created when the user calls. With Echo II, no recorders are involved—just chips and software.

To fool with the Echo, you just boot it up and type in words; your Apple says what you write. Well, almost what you write. Some words just aren't spoken the way they're spelled; in many of these cases, the words are part of the Echo's vocabulary, but, in others, they aren't and you must spell somewhat phonetically. For example, Echo can't say its own name as spelled; it'll say "eh-cho"; but *eko* does the trick. Yet Echo has no problem with a word such as *synthesizer.* Anyway, half the fun is getting Echo to say things correctly. Actually, by the time you read this, no one will have to use trial and error to ensure Echo's pronunciation. New documentation, not available during the testing time, shows exactly how to write every sound.

Perhaps you're imagining the Echo II as sounding like the robots on "Battlestar Galactica," low-voiced and monotonal.

Echo does speak in a deep baritone, and it's capable of sounding exactly like those robots—or like Mickey Mouse, if you like; but its normal voice is anything but monotone. It's richly modulated in sixteen different levels. When you use the Echo in programs, you can call for any word or sound in several configurations of speed, volume, and pitch.

By mid June, the baritone Echo will have a counterpart snug on the same card: a female voice. Evidently, it's much harder to simulate a female voice accurately. You're apt to get all sorts of strange sounds, the best of which is a male falsetto. But Street has found the way, and Echo will have a true female voice synthesis at that time.

Echo II takes much of its technology by license from TI. Remember that radio ad campaign: "This is Hal, the talking computer. . . ." That—or what that was advertising—was TI's voice synthesizer. Milo Street has made many revisions and refinements in the basic engineering, however, and the end result is a super product.

Echo II has shortcomings. Perhaps you won't understand all of what it says at first; and, while you'll be on to it in very short order, the fact remains that some of the sounds are not perfected. However, updates, several of which have occurred during the review period alone, will be available to those who have bought early Echos.

At least one game already exists using the Echo II—and requiring it for playing the game with sound: that's *Bingo* from Continental Software (Marketalk Reviews, May); and more are planned.

Eventually, the Echo's major usefulness is apt to be found in different areas. Imagine an orally disabled person, already helped by the computer by being able to type messages on the screen. With the Echo, that person has a voice. Blind people, now using talking programs that take up much of their Apple memory and disk space, will be able to hear their computers with next to no memory loss.

This raises the chief practical difference between the Echo II and taped voice generators: space. Echo only uses ten to twenty bytes per word.

To some of us, however, the major difference is the difference itself: that, with the Echo II, the Apple is generating the voice without any human voice's help.

That's the line they're giving us, anyway. We all know the truth: It's really Maxwell, come to life and making his home in our computers. MCT

Echo II Voice Synthesizer by Milo Street, Street Electronics, Anaheim, CA. Applesoft, disk. $225.

Cartels & Cutthroats. By Dan Bunten. Fully faithful to its brand name yet in an entirely different vein from any of that company's previous games is Strategic Simulations's game of business strategy, *Cartels & Cutthroats.*

Resembling a bookshelf game of the pre-Apple era, the divertissement gains color and liveliness in its translation to electronic media. It also gains players: the computer will play for as many as five of the game limit of six players. When playing three and four roles, the Apple diversifies its strategies widely, to the point that its companies are far apart in the final toting up. Of course, the game can also be played by six people, or two—although this is a good time to bring in Apple.

The thrust of the game is that you play the role of president of a company for periods from two to ten years' duration. You buy raw materials, produce goods, set prices, budget advertising and research and development, negotiate and pay loans, build or sell factories, and listen to and direct your top staff, all toward the purpose of raising your company's equity position.

Turns are quarterly planning sessions, during which you look at reports, confer (via their memos) with your staff, determine your course of action. All the time, you must keep in mind the state of the economy, which may be rising or falling and may be wildly or mildly inflated, among other factors.

In a given game, all businesses have product lines in the same category, which may be luxury goods, necessities, or mixed goods. Each brings unique situations and calls for different strategies. Product line and general economic outlook are generated by the computer at the beginning of each game, but if you don't like what the Apple first offers you, you can have it generate a new set of given circumstances until you find one you like. Chance influences the game to various degrees; how important it will be in an individual game is announced as part of the opening scenario. Chance happenings, such as general transport strikes or government levies, affect everyone; each company's ability to cope with an event is determined by its president's prior planning. Labor strikes that affect individual companies are directly attributable to the player's choices.

There are four configurations for gameplaying: open game, where everyone watches everyone's reports and plays (except the computer, who keeps its business to itself); closed game, where players look at reports and make plays privately without other players looking on; hard-copy game, in which all reports are printed out for studying and planning; and the beginner's game, limited to eight quarters and luxury goods for simplicity and learning. Beginners' reports, which give tips on how you're doing and how to improve, are available in the other games as well.

Between quarters, there is a colorful animated graphic of the world of industry, followed by a pithy, if tongue-in-cheek, business aphorism. At this point, also, is the best opportunity to save your game.

There is nothing to stop several players from joining in all sorts of anti-antitrust schemes; the good old government plays no part in the game, except to raise taxes now and then.

Dan Bunten, whose excellent programs are popping up throughout Appledom (his previous SSI program is *Computer Quarterback),* has done an excellent job of enlivening the business world in computer game form; although necessarily simplified in detail, the game is carefully planned to simulate real-life results for each action, given each circumstance.

As we've come to expect from Strategic Simulations, the game is beautifully packaged with detailed glossy game manual, game card, and a pad of "Business Planning Sheets."

Although *Cartels & Cutthroats* is not a game that will have you doubled over in fun and laughter—except, possibly, if you're playing with a lot of very cutthroat friends—it will keep you going back for more for the satisfaction of seeing what will happen if you play it just a little bit bolder or. . . . MCT

Cartels & Cutthroats by Dan Bunten, Strategic Simulations, Mountain View, CA. 48K, ROM Applesoft, DOS 3.2 or 3.3. $49.95.

Olympic Decathlon. By Timothy W. Smith. Even people who take little regular interest in sports often feel a thrill upon hearing the opening strains of ABC's Olympics coverage theme and seeing the nations of the world, as represented by their finest amateur athletes, join in parade. If you're of this mind, Microsoft's *Olympic Decathlon* is directed to you as well as to sports buffs.

Olympic Decathlon, upon booting, comes as close to purveying mood of Olympic pomp and ceremony as a computer program probably could. An athlete pulling a title word runs onscreen, reminiscent of the Olympic torch runner. He is not reminiscent of any hi-res character you've seen on the Apple before, however. He moves in total animation, seemingly anatomically correct in each of the many motions that make up running. Another athlete puts the shot to dot the i in Olympic. Then a circle appears to be lowered from the ceiling, like a flag, and spreads and lowers some more to form the Olympic rings.

Only then does sound occur: indeed, it is the ABC theme we've all heard innumerable times. When the subtheme begins, the screen switches to an introduction, with a message to hit the escape key to go on. But hitting it will avail you nothing until you have listened to the last strain of the melody.

What's in store then is one of the most unique game programs ever for the Apple.

Decathlon presents the series of ten competitive athletic events that make up the modern version of the ancient Greek contest for overall excellence in track and field activities. The required abilities had a much greater applicability to life then, but the admirability of their achievement cannot be overlooked even today.

Decathlon athletes must compete in all ten events during the span of two days. The events are the 100-meter dash, the long jump, the shot put, the high jump, the 400-meter dash, the 110-meter hurdles, the discus throw, the pole vault, the javelin throw, and the 1,500-meter run.

This is precisely the content of the program. As many as six people may compete through the ten events with each other and with the record-breaking score achieved by Bruce Jenner at the real 1976 Olympics. The latter is detailed in the instruction book.

There is a large measure of unreality inherent in trying to simulate athletic contests on the computer keyboard, but, to the extent possible, Smith has done it. Running, where running is a major focus, is usually simulated by two fingers on the keys. This is perhaps the least accurate of the simulations. All the rest concentrate on the timing involved in executing the event and on the judgment of relative position. Whether or not these are accurate simulations of these factors as involved in each sport is difficult to say. That these skills are required to play *Olympic Decathlon* is a fact. Each event is unique, and you'll find that each person you play with will probably excel in a different area.

Although *Decathlon* is not addictive in the ways that *Super Invader* and backgammon are, it presents a pleasant challenge that will draw you back to the game many times. Outstanding graphics, animated athletes that look and move more like real people than almost any before (it's hard to tell about Maxwell—he has so many more clothes on than the athletes) add to the game's enjoyability. Particularly of note in this regard, besides the opening people, are the hurdler and, in his jumping position (the Fosbury Flop), the high jumper.

Decathlon also offers a practice mode, in which you can concentrate on the events of your choice. MCT

Olympic Decathlon by Timothy W. Smith, Microsoft Consumer Products, Bellevue, WA. 48K, disk, paddles. $24.95.

Zork. By Marc S. Blank, Timothy Anderson, Bruce Daniels, P. D. Leblins, Scott Cutler, and Joel Barez. Widely heralded as the adventure to beat the original *Adventure*, first published by Microsoft, *Zork* gives living up to its reputation a good stab. Unfortunately, for most people, the Microsoft *Adventure* (or the Apple *Adventure*—they're essentially the same thing) was for thousands of people their introduction to the genre of computer adventuring. It would take something far superior to the original *Adventure* to overpower the delight of first-timing that thoroughly enhances our memories of the two-pit room, fee fie foe fee, and the gentle faithful bear.

Strictly as an adventure, *Zork* seems right on a par with the original; in fact, it appears to be somewhat more complex, and it adds some flavor with clues and possessions that your own general knowledge can help to manipulate. For instance, when you come to an altar lit with candles upon which lie a dinner bell and an open Bible, what comes to mind? If it's Shakespeare's *King John*, you're just that much farther ahead.

On the other hand, if some of the locations seem a bit contrived, the great number of them makes up for that. The mere fact that in *Zork* you have at last a second real, logical, epic adventure to conquer makes pale any objection to the similarity of tone and atmosphere to those of the original *Adventure*. Like that program, *Zork* was originated on mainframe computers.

One clue: Don't always assume defeat just because you're dead.

In its execution of the no-res computer logic game form, *Zork* surpasses original *Adventure* with one giant leap. Until *Zork*, all adventures—except the uncategorizable *Prisoner*, which stands alone—limited the player to two-word input, occasionally allowing a qualifying phrase in a second command; for example, you say, "Kill fiend." Adventures respond with, "With what?" You answer, "With Battleax." And the computer, more often than not, says, "Battleax has no effect." In contrast, in *Zork*, you can not only say, "Kill fiend with battleax," whenever you like, you can actually tell it, "Attack the lousy fiend with the battleax and kill the fiend with the sword and take the treasure."

Of course, long sentences don't get you any favors. *Zork* will probably answer, "Battleax won't work. You missed. The fiend won't let you," in response to your three commands in order, but it's much more satisfying to write them that way. One situation in which this facility is bound to prevent lots of high blood pressure occurs when you want to combine manipulations of things. For example, you have an empty box and you have a small treasure. Maybe you could carry more if you could put the treasure in the box, you reason. No way, in previous adventures. But, in *Zork*, just say, "Put the treasure in the box." It *won't* make any difference to what you can carry, but you'll feel a whole lot better.

Although, much of the time, you'll find yourself using the familiar two-word commands anyway, the ability of *Zork* to understand long, compound (not complex) sentences, and even some questions, adds a new dimension of pleasure to the genre. MCT

Zork by Marc S. Blank et al. of Infocom, Personal Software, Sunnyvale, CA. 48K, disk. $39.95.

Oo Topos: An Extraterrestrial Adventure. By Michael Berlyn, Sentient Software (Aspen, CO). "Five parsecs from the Galactic Council's space station, an insane alien race took control of your ship and forced it down on their planet. They stripped your ship of cargo, and then they stripped your ship. All you have to do is find the serum the council gave you and the rest of the cargo the aliens stole. . . . You're in a prison cell. By constantly kicking the auto-lock mechanism, you've managed to free the door. Through the barred east window, you can see the stripped-down hull of your ship on the beach of a pale-green sea."

The introduction and opening description from *Oo Topos* set the mood and tenor of the first real contender for adventure laurels written directly for the microcomputer. You do not have to travel far in this no-res but highly graphic game to know that you're keeping company with a peer of the original

Adventure and *Zork*. No trivial puzzle, this. No production line rearrangement of standard parts here; this is handmade and hand-polished in every one of its multitudinous details.

Michael Berlyn is a writer of science fiction. He bought an Apple to process the words he invented. Pretty soon, he found his pleasure centering more on the using of his Apple than on the writing he was using it for. So he began to program; and, naturally, rather than abandon his early love, he brought his fiction to his programs.

The result is an original science fiction tale, complete with plot, atmosphere, word paintings, feelings, in the form of a computer adventure everyone can take part in. The writer's skill shows through consistently. Thus, an alien is not a little dwarf who drops an axe, but a "silent creature a meter high" who, "startled at seeing you here, drops his laser and streaks around the corner"; but who, when he seeks you out and you shoot at him and miss, "giggles wildly, doubling up in glee."

Berlyn must have been no hack writer, but one who did the homework of his trade. His research in the world of science and science fiction serves us in the objects we find and use during *Oo Topos* and in the buildings, spaceships, and planets we must find our way through.

Oo Topos is on the scale and logic level of *Adventure*. Unlike *Zork*, this is where the resemblance ends. Everything else about *Oo Topos* is totally different—except, of course, that apparently mandatory twisting maze. *Oo Topos*, however, has several mazes that are quite different. One, set in a jungle, is so realistically described that you don't realize you're even in a maze until you note that you haven't really gotten anywhere in the last half-hour or so.

Oo Topos requires two-word commands, but it won't settle for nonsense. If you tell it, "Go door," it says, "That doesn't make any sense." And, of course, it doesn't, much. In little ways, the program is very player friendly. In this game, if you're carrying a full load including goggles and gloves, and you want to pick up something else, go ahead and wear the goggles or gloves. Doing so will enable you to carry something else. On the other hand, if you're carrying something and you cause it to be heavier—feed a live animal, for example—be prepared to leave behind the next item you drop, because you're overweight and won't be able to pick it up again.

If you're about to do something fatal unnecessarily, *Oo Topos* won't let you; "You can't go east," it will say, "those force fields would shred us like paper!" And commands thwarted because of the program's vocabulary limitations don't count a whit in *Oo Topos*'s system of time. No creature can get you because you guessed the wrong word.

That *Oo Topos* was originated and written on the Apple for the Apple is no small feat. For months, in explaining the gap between the original *Adventure* and others of the genre, people would drop their voices knowingly and say, "Well, of course, *Adventure* was originally written on a mainframe." Today, they're giving the same explanation for *Zork*. Such people will have to find a new explanation for *Oo Topos*.

Despite its lack of fanfare, such as *Zork* had, and despite its brand-new company, *Oo Topos* appears to deserve a top spot in the realm of adventuring. 48K, Applesoft, disk. $29.95. MCT

Oo Topos: An Extraterrestrial Adventure by Michael Berlyn, Sentient Software, Aspen, CO. 48K, Applesoft, disk. $29.95.

DOS Boss. By Bert Kersey and Jack Cassidy. Ever stumble over a log when typing catalog? Or unintentionally delte an e in *delete*? If so, those phunny pholks at Beagle Bros. have a program for you.

DOS Boss takes you inside your operating system, where you can rearrange the commands to your fingers' delight. Now you can turn catalog into CAT, INV, or whatever you please, subject to length constraints.

Confound your friends! If you want to keep snoopers out of your program list, turn list into lust and don't tell. If you want to stop them from swiping your programs, change save to keep and read to save. When they try to save your work on their disk, they'll get a nasty error message.

You can make it really nasty if you like, because *DOS Boss* gives you access to the error messages as well.

This handy utility also lets you reformat your catalog into two or four columns, so it won't scroll off the screen. If that cramps your style because your program names are long, you can conserve some space by getting rid of the lock symbol, language code, and sector size. You can even restructure or eliminate the volume header.

Once you've got DOS scrambled to your satisfaction, there are two ways to put it on disk. For a new disk, simply type in your hello program and init; your new DOS is there forevermore. For disks already initialized, *DOS Boss* has a utility that will append all the appropriate pokes onto any Basic program. Whenever you run that program you'll plant your new DOS in memory.

One of the things you'll probably appreciate most about *DOS Boss* is the amount of information that comes with it. A thirty-six page booklet not only provides details of the individual pokes required to customize your operating system, but even includes the program listing of *DOS Boss*.

There's a whole lot of other goodies in the booklet as well—handy information about both Basics and DOS, about Apple graphics, and so on. Some of this stuff is of real practical value, and some of it—like a four-line program to make your typed input start at the lower right corner of the screen and progress backward—is just plain bananas. It's all done with great wit and style. CS

DOS Boss by Bert Kersey and Jack Cassidy. Beagle Bros., San Diego, CA. 32K, Applesoft, DOS 3.2 or 3.3. $24.

Snoggle. By Jun Wada, Broderbund Software (Eugene, OR). One of the more curious aberrations of contemporary programming was displayed by the original, preproduction model of *Snoggle*—at the time called *Puckman* after the arcade game of the same name. *Puckman* played sideways.

This was not too serious an impediment to enjoyment if you were playing with a monitor that could easily be turned on its side to give you a proper, vertically oriented picture. But if you were using a twenty-seven inch console or a six foot big screen as your display device, only a double-jointed back would permit you to play the game the way it was meant to be played.

Ordinarily, one might suspect that such inattention to detail as determining the appropriate axis on which to base the game would spell a quick and deserved death to any piece of software. But *Snoggle* doesn't appear to be just another arcade pastime. It seems to have captured the imaginations of the arcade set; the corrected version is selling like hotcakes.

Perhaps it's a masochistic streak coming to the fore. Broderbund claims the program has eight levels of difficulty. But the only way to get past the easy level is to complete it, at which time the program promotes you to the second difficulty level. It's not likely that you'll ever see the other six alleged levels.

The essence of the game is a maze, the pathways through which are filled with dots. The player attempts to eat all the dots while avoiding four ghosts that roam the maze. Four of the dots, one near each corner, give your puck the ability to swallow the ghosts for a short period of time, after which they reincarnate in the center of the maze and start after you again. If you can swallow all the dots, you'll get promoted. Points are given for each dot swallowed and for each ghost done in.

The instructions say that if you reach a total of ten thousand points, you'll get another puck. That occurrence is about as probable as your winning a *Reader's Digest* sweepstakes.

But arcade fans are a persistent lot and one suspects that *Snoggle* would have achieved a high sales level even if Wada had programmed it upside down. ART

Snoggle, by Jun Wada, Broderbund Software, Eugene, OR. 48K, $24.95.

Math Tutor and Spelling Tutor. By Ed Magnin. These are educational programs designed to be used in the home to supplement schoolwork. Their major, if not only, function is to provide custom drill exercises kids can use to memorize or gain speed in areas requiring those abilities.

When students first boot the disk and choose a tutor, they must enter the date and their names. The computer then seeks their files or makes them new ones, and the drill begins.

The *Math Tutor* offers drill in addition, subtraction, multiplication, and division, and it allows the student, parent, or teacher to input parameters within which the work will fall. For example, you might choose numbers up to 1000 for all parts of an addition problem, up to 100 for multiplicands but only to 12 for multipliers, and up to 1000 for dividends but only up to 100 for divisers. Then, you choose what combination of the math functions you wish to be used in the drill. Here you can specify addition only, multiplication and division, or all, for example. Finally, you input the number of seconds to be allowed during which an answer can still be considered correct. Until you or the child choose to change them, these parameters will obtain every time the child they were designed for signs in with the computer.

The computer dislikes wrong answers so much that it erases them on sight; then it waits for the student to try again. It only goes on after the right answer is given.

After the drill—there is no limit to the number of problems; the child merely enters Q to quit—the session results are saved. They may also be printed out at this point if you wish. Results consist of a series of dates on which the particular child used the program, the number of problems of each type the child attempted, and the number of problems of each type the child did not get right within the time limit.

The *Spelling Tutor* works exactly the same way logistically but, instead of parameters, the computer asks for a spelling list of up to fifteen words. Then it flashes these words on the screen one at a time, and the student must type them as soon as they disappear. Each time the answer is wrong, the computer puts the correct word on the screen again, for a slightly longer time. The computer tolerates four wrong answers here; then it leaves the word on the screen and waits for the student to type it directly underneath while looking at it.

In terms of space, the program has provided for all possible lengths of words in English. It would even accept antidisestablishmentarianism. But it's not too hyped on slang or Julie

Andrews; it refused to take supercalifragilisticexpialidocious.

The timing factor and simplicity are the major assets of the *Tutors.* Simplicity is evident in the easy entry of customized data and in the absense of complicated instructions and any beyond the most necessary bells and whistles.

The real-time timer puts kids under pressure to make their knowledge work for them. For example, in math, children must either space to the right, then backspace, put in answers right to left as they would on paper, or they must learn to take appropriate shortcuts, estimating and revising as they go.

This is the timed drill program available, as versatile and as easy to use as any and that alone makes it worth the price of admission—unless you believe people are just as well off depending on pocket calculators throughout their lives. MCT

Math Tutor and Spelling Tutor by Ed Magnin, Telephone Software Connection, Torrance, CA. 48K, disk. Prices by modem: *Spelling Tutor,* $20; *Math Tutor,* $25. Over the counter, $25 and $30, respectively.

The Complete Mailing Label and Filing System, with Supplement and Adjunct. By Don Jones. As is true of every mailing list and many other business utility programs, this is merely a focussed data base. What distinguishes one from another are the details of that focus—format of files, ease of use, ease of editing, print abilities, utilities provided, and so on. Usually, you get what you pay for; the more a program will do, the more you pay.

The people at Avant-Garde Creations are philosophers who learned to program. Their way of thinking is not at present in the mainstream. But it is a very benevolent point of view, and, while it doesn't appear that they shun profits, they believe in pricing their products at a certain amount over cost; they do not believe in charging whatever the market will bear. Consequently, when Avant-Garde offers a superior product, it is a hell of a bargain.

The Mailing Label and Filing System is exactly that. It is not the package for absolutely every business, although it uses the honor system for protection just so those who buy it can customize it to their needs if they like; but it will fill the bill for many small businesses, for specific needs in larger businesses, or in the home.

The foundation of the package is the basic system, which may be used alone. It must be used to create all files and input all basic data for the package. It can produce mailing labels in the format you wish. A unique feature is a customized line you may choose to put at the beginning of your labels; it must be the same line for all labels printed in a batch, but it can say whatever you want, and it can say it in graphic letters. For your convenience, the disk contains Superfont, one of Avant-Garde's graphics printing utilities, which enables you to choose from among eight type styles in any of nine sizes.

When you create a file, you must specify which fields from a list of eighteen you'll be using in each record. These include personal names or business name as well as contact and contact position, balance due to the subject, and balance the subject owes you. In addition, there are three free fields that allow up to fifteen characters each for comments, and there are three single-character code fields, marked for Active Account Code, Credit Rating Code, and Business Prospect Code.

The records you create here can be manipulated in many ways, but once the record format is set, it stays with the file forever. With the offspring program, the screen and print formats can change infinitely.

Full editing, deletion with fill in, and search and view utilities are included. Each file, or disk, can contain from 477 to 983 records, depending on format.

The second part of the trilogy is the *Mailing Label and Filing System Supplement.* This program requires the main system. The *Supplement* contains two disks. One presents numerous utilities for manipulating the data on your mailing list while using an alphabetical sort; the other does all the same things while using your list in zip code order.

What the *Supplement* enables are variously formatted lists, such as phone lists; contact lists; lists of debtors or creditors with amount due or owed plus totals; labels for partial lists, such as only for those who owe money; and multiple sorted lists or labels, where, for example, you need only those people who buy more than a certain amount of your product and owe you less than a certain amount of money and like you to call them twice a week.

Alphabetical customized reports are enabled by the *Supplement.* This program allows you to create titles and number report pages, label columns of data, specify field to be printed, and choose tabs for the columns. You can make reports on all records in the file, or you can choose only those that meet certain criteria. For greater convenience, the program memorizes the records chosen by those criteria, and you can further manipulate those records with a second count-and-sort search.

The third part in this system is relatively specialized. Again, it requires the basic system. Called the *Mailing Label and Filing System Adjunct,* this program allows you to do partial field sorting, using as little as one character or code per field, on your data. You can sort on as many as forty-five single-character number or letter codes at the same time as you sort on as many as three ranges and eighteen other variables with as many as nine values each.

The *Adjunct* makes the system ideal for statistical studies, questionnaire response compilation, or contest results, for example. You can then choose entries for labels on this basis and have the labels printed in alphabetical or zip code order; you can print out similarly chosen phone lists and reports.

If you don't like the labels on the fields in the system for a particular purpose, you're stuck with them, unless you're willing to go into the program itself and change them—Avant-Garde won't mind. They care that their programs are usable for you, and they do their best to make them so. Because of this attitude, updates to Avant-Garde programs are easy and immediate: for example, latest editions of the system carry a change you can make to allow nine-digit zip codes. The system already provides for foreign zips.

Also included with the system is a dual-drive copy program. The disks come in DOS 3.2, but can be Muffined.

The most remarkable part follows. MCT

The Mailing Label and Filing System, Supplement, Adjunct by Don Jones, Avant-Garde Creations, Eugene, OR. 48K, ROM Applesoft, disk. Individually, the *Mailing Label and Filing System* and the *Supplement* each sell for $24.95, the *Adjunct* for $19.95; all three, $59.95.

Impressions

□ **Space Raiders.** By Paul Lutus, United Software of America (New York, NY). Lotsa hoopla on this program. Your Apple splatters stars all over the screen and screams its little speaker out as you make the hyperjump to your chosen space coordinates. Once there, though, the game is simple target practice, your objective being to maneuver aliens in front of your crosshairs and blast them out of the cosmos. They're firing on you, too, of course, so don't forget to put up your shield. If the enemy knocks out your engines or weapons, you may still be able to limp back to a nearby starbase for a fix. Control of your ship is through either joystick or the IJKM diamond. If you're using the diamond, you'll work up some nimble fingers; that alien moves around pretty quick. 48K, disk. $29.95.

□ **Black Hole.** By David Durkee, Dynacomp (Buffalo, NY). If you like black holes, you can really get into this one. Besides being a challenging game, the program is a fascinating lesson in gravitational physics. The object is to fly an orbit around a black hole without getting sucked in. A realtime clock counts the amount of time you spend within a certain radius of the hole; if you can spend as much as two minutes there, you win. You lose if you stray too far from the hole or, naturally, if you succumb to its gravitational pull. To control your course you may change the attitude of your ship in forty-five-degree in-

GOTO 65

Mind Your Business

BY PETER OLIVIERI

One of the most frequent questions business people have asked since the introduction of the Apple has been, "How can I use the Apple in my business?"

To meet this need, Mind Your Business is devoted to providing you with ways the Apple can help you in the day-to-day activities of your business.

Whether you are a participant in a small, family-owned enterprise or an executive with a larger company, hoping to use your desktop computer to do some financial forecasting or modeling, this column should be of interest to you.

Each month, the column will contain three sections: Section 1 will be a *tutorial* on a business application. In this section, you'll find discussions of computer applications in the functional areas of business (finance, marketing, accounting, production, and so on) or, perhaps, explanations of how *VisiCalc* or another tool might be used to forecast sales or calculate useful financial ratios.

The intent in this part is twofold. First, a particular management technique such as forecasting will be described and, second, an illustration of how the Apple can help you implement the technique will be presented. Some of the topics we'll recover in these tutorials include:

Determining the computer needs of your business
Where to start first
Using the Apple for planning
Using the Apple for control
Using the Apple for decision making
Managing the data needs of the business
Creating an effective management information system
Providing adequate backup for your data
Planning for growth
Pitfalls to avoid
Evaluating software
Designing business forms that work

A valuable by-product of this section will be a compilation of reference books on computers and business that may be appropriate for your bookshelf.

Section 2 of Mind Your Business will review business software. If the question businesspeople most frequently ask is about how to use the Apple in a business situation, then a close second asks how to know if a particular business package is any good. Indeed, there are many business application packages on the market. Each has its own strengths and limitations. This review section will detail these within an objective evaluation of the package. When it is appropriate, ways in which that particular package might be used in your business will be detailed.

With several data base management systems, accounts receivable systems, and mailing list programs currently on the market, it is almost impossible for businesspeople with Apples to know which system is right for them. This section of the column will give these products a rating for their ease of use, documentation, adaptability, and reliability.

Among the packages to be reviewed are:

data base management systems
accounts receivable systems
mailing list programs
word processing programs
tax planners
financial analysis packages
general ledgers
forecasting programs
inventory control packages
loan analysis packages
graphics systems

The third section of the column will respond to questions sent in by readers. Here, you can ask more specific questions about your particular business or application. Your questions may involve the application of a management technique, the use of a certain program, or perhaps the selection of the appropriate hardware for your purposes to complete your computer system. This will be your opportunity to get some personal feedback.

In fact, this opening column presents our first reader question, sent in by J.F. from Woburn, Illinois.

I am seeing more and more advertisements for data base management systems. What exactly is a data base?

Every business needs to maintain data or files of information. These data may be in filing cabinets, on index cards, or in the head of the owner of the business. Since a computer can remember things, businesses with computers store this data on some storage device that is connected to the machine.

To be efficient for processing, the data must be organized in some way. A *data base* is the organized collection of data about a business. This data base may be made up of one or more *files*. For example, the business may maintain a personnel file, a customer file, an inventory file, and an accounts receivable file.

Within each file, there are *records*. A record is a set of information about one item in a file. Thus, a customer record might contain the name, address, and telephone number of a particular customer.

Within each record, there are *fields* of information about that record. One field of information about a customer would be the customer's name.

Finally, each field is made up of *characters*. At the lowest level, the data base is made up of letters and numbers.

The data base management system that you read so much about is a set of programs designed to help manage all of this data. It provides for creating files, adding records to a file, deleting records from a file, and changing the contents of specific fields. Most systems also allow the user to create certain useful management reports.

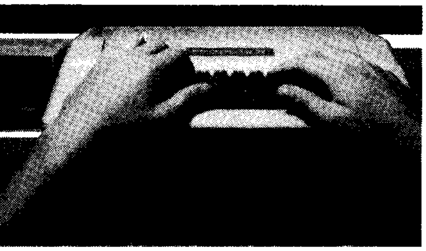
We will be reviewing various data base management systems in future columns as well as providing tutorials about the use of such systems in a business setting.

Please address all questions to Mind Your Business, Softalk, 11021 Magnolia Boulevard, North Hollywood, CA 91601.

The phrase, "Mind Your Business," appeared as a slogan on coins minted by the Continental Congress. That august body, founded after the colonies declared themselves independent of England in 1776, produced the Constitution and resulted in the formation of the United States in 1789.

The slogan appeared as part of a legend, all of which translates as "I, time, fly; therefore, mind your business." The phrase represented the responsible free enterprise philosophy that was America's unique combination of the work ethic and individual freedom. Believed to have been coined by Ben Franklin, the phrase was meant as a serious and benevolent reminder among colleagues.

TRADETALK



In the microcomputer industry, change is the norm as technological advances make yesterday's advanced products today's state of the art and today's state of the art tomorrow's primitive technology. In the ten months that *Softalk* has been published, this column has made an effort to chart the growth of the companies and individuals in the industry.

□ For the first time Tradetalk gets the opportunity to take cognizance of an executive getting her second promotion in that short time frame. **Tamisie Honey,** whose promotion to product manager at **Advanced Business Technology**, was noted in October's issue, has now been named general manager of the company by president **James Tennyson.**

□ Another October Tradetalk item revealed the sale of **Programma** (Burbank, CA) to **Hayden Book Company** (Rochelle Park, NJ) and the appointment of **Dave Gordon** as vice-president and general manager of Programma. Now Gordon, whose massive publishing efforts early in the development of the personal computer supported maturation of the marketplace, has been replaced as general manager of Programma by **Mel Norell.** Norell, who has been heading Sigma Systems, another Hayden subsidiary, was one of Programma's founders along with Gordon.

□ Six months ago a call to **On-Line Systems** in Coarsegold, CA, reached either **Ken** or **Roberta Williams,** who were not only the owners, but also the only employees of the company. Now a staff in excess of ten has been added in support of the expanded company product line. Latest additions are **Ed Peters,** ad manager; **Dan Doyland,** administration manager; and programmer **Drew Harrington.** Peters is, like the Williamses, a transplanted southern Californian. He brings a diverse background to his position and replaces John Williams, who has assumed other duties. Doyland has a degree in agricultural business with graduate work in accounting with a computer emphasis. Harrington, a native of Hayward, CA, has three years of programming experience on mainframes as well as on the Apple. He's pursuing a degree in computer science with emphasis on artificial intelligence. Meanwhile, Ken Williams has fired himself from the On-Line staff in hopes that he'll be able to get some programming done. His efforts in that direction were interrupted by a gala celebration of the company's first anniversary in mid May. Numerous friends and business associates were housed near Coursegold for the weekend, compliments of On-Line, so they could attend the western themed dinner and dance.

□ Last month, *Softalk* spotlighted **Edu-Ware** (Canoga Park, CA) and showed pictures of the staff and their new facility. In regard to the new facility, the article is already outdated. Expansion has caused the company to take an additional thousand square feet on the second floor of their present office building. **Sherwin Steffen, David Mullich,** and the rest of the programming crew will occupy the high ground, while **Steve Pederson, Mike Leiberman,** and **Wendy Peterson** keep the old offices along with expanded storage and shipping facilities.

□ **Dave Wagman,** president of **Softsel** (Marina del Rey, CA), announced that

GOTO 79

Exec Apple: Mike Markkula

from page 6

tendency to incur bent pins in the process. Because the new sockets sit higher, contact with the bent pin is still possible under manufacturing conditions and the unit will test out correctly. Only under the stress of shipping and handling will the contact be broken, leaving the retailer and end user with a diagnostic headache, but one that's at least readily and permanently solvable.

At the present time, Apple is replacing all early motherboards that have the less reliable sockets with new motherboards when they are returned for repair. A definitive policy regarding upgrading these early IIIs is in the works.

The socket problem also wreaked marketing havoc. Apple decided to ship dealers a demo III in advance of shipment of most units to give the retailers time to familiarize themselves with the machine. Many retailers who tested the III got the impression that it wasn't up to the quality of the II—an impression that lingers today.

Markkula recognizes the source of their disenchantment, but it's nevertheless a source of disappointment that some dealers won't give the upgraded version a fair chance. In his view, the III is today as fully reliable as the II.

He also looks for the advent of significant software support for the III by the end of the summer. Pointing out that such support would only follow when there was a sufficient volume of computers in the field, Markkula said the III was now shipping at a volume that should interest software publishers. Apple has given seminars for and provides information to those publishers who have made tentative feelers toward supporting the III.

In addition, Apple hopes to have additional software ready for marketing in that same approximate time frame.

A Man with a Bike. Markkula cites industry gurus who believe that what is today approximately a one-billion-dollar personal computer market will balloon to five or six billion dollars in the next four years. But he further suggests that the successful introductory phase of personal computing is an ongoing effort that may take until 1990 to complete.

He sees no limit to the uses and applications of the personal computer. Markkula likes to tell the tale of the anthropologist who made motion efficiency studies of several members of the animal kingdom. Man stood well down the list. The same study, however, measuring man with a bicycle, showed man as the most efficient animal.

The analogy is to man's ability to enhance his inherent ability with tools. Markkula views the personal computer as the first general purpose tool devised to enhance man's intellect. And, in his view, the computer can go with man wherever man's intellect can conceive to go. In light of man's inability to chart finite boundaries to the intellect, it appears that those boundaries might then be equally as limitless for the personal computer.

Markkula's business plan has two long-range items still pending. He wants to build a solid company that will still be here in the year 2000 and he's looking for Apple to crack the Fortune 500 in this decade.

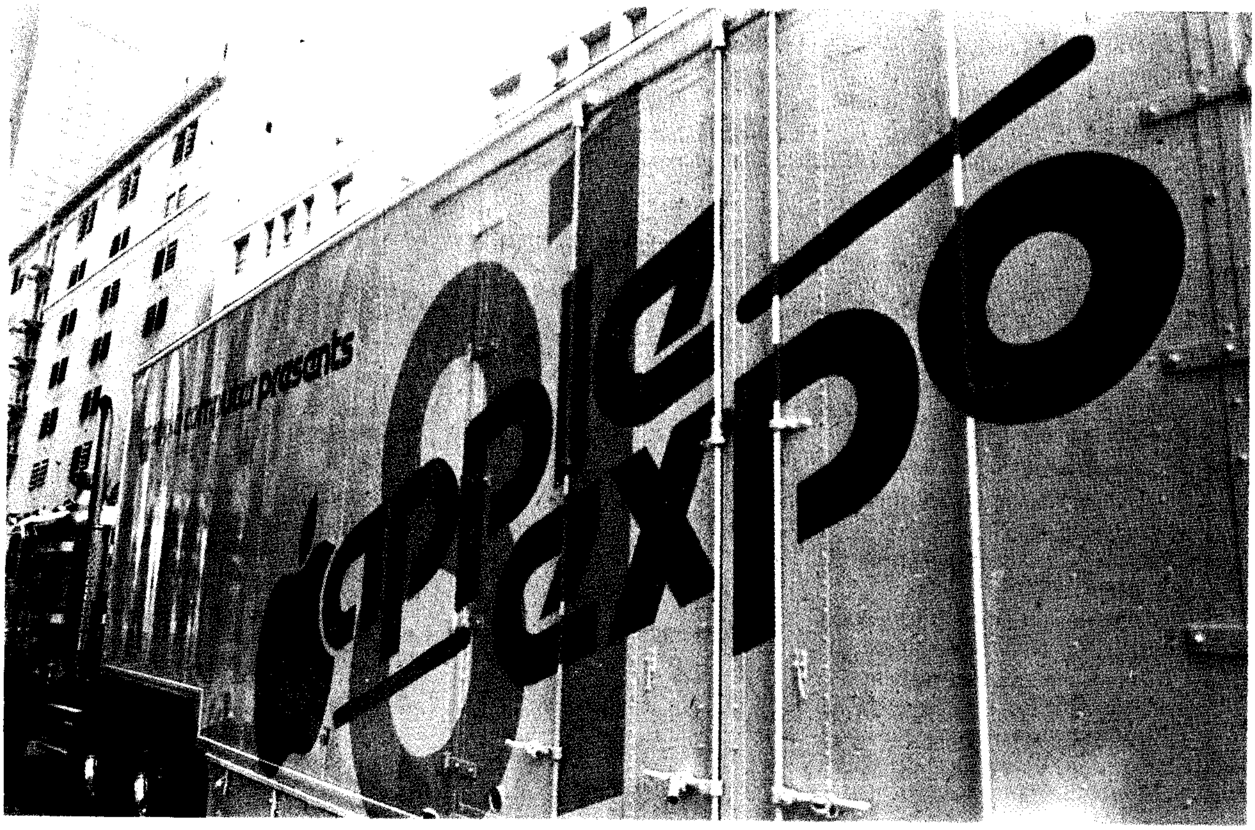
When those objectives seem secure, he may opt for re-retirement. Then, who knows, maybe somewhere out there he'll find two sharp kids building a personal space shuttle in their garage. . . .

From the top: Unloading Expo materials outside the Biltmore Hotel in downtown Los Angeles. Apple handled the logistics of building, dismantling, and transporting Expo materials for all the exhibitors. Steve Grawd and Bridgeman Carney manned the Anadex booth. Left: Las Vegas retailer Ike Jordan attended the Los Angeles stop of the Expo. Right: Phil Roybal, Apple's manager of editorial services, introduced Mike Markkula at the opening luncheon of the Los Angeles edition of the expo. Ken Klein and Lou Long of Stoneware explained DB Master.

from page 24

that many of the companies who found it most profitable to go on the road with Apple were ones with new or technically sophisticated products that needed that kind of dealer approach to start making inroads on the marketplace.

The almost unanimous view of the exhibitors was that the Expo was invaluable in reaching Apple dealers, but the reviews of the public session were more mixed. One of those on the pro side was James Tennyson, president of Advanced Business Technology, who found feedback from the user public valuable as the time he was able to devote to retailers.



Teach your little Apple to talk to Western Union Telex and TWX

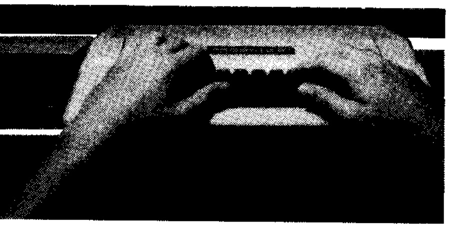
From the top, left to right: A caravan of these eighteen wheelers made an impressive sight on the freeways with their Apple logos. James Tennyson, president of Advanced Business Technology, displayed his keypads and bar coders. James Dow, president of Microcom, was kept busy explaining his products to make the Apple a stronger communicating device. E. E. Campbell displayed MPI's new dot matrix printer with graphics dump. Printer is reputedly one of the few wholly made in the United States.

Softalk photos

Logistics for the mobile exposition were handled exclusively by Apple. All exhibitors shipped their Expo materials to southern California, where a caravan of eighteen-wheelers was mounted to traverse the country. Apple also handled construction and dismantling of booths at each site.

The Expo lacked the excitement of the Faire in that most of the products were both more esoteric and better known than many at the Faire. On the other hand, there was more opportunity to delve in depth into those products in which a guest was interested. ⊐■

MARKETALK



Impressions

from page 58

crements and employ varying amounts of thrust. Fuel is limited, however, and if you run out you're left to the mercy of nature. Gauges below the hi-res screen show you such vital data as your ship's velocity and distance from the hole, the current gravitic force, the amount of fuel remaining, and the length of time spent so far in the target area.

Once you've reached the two-minute goal without falling into oblivion, the object of the game becomes winning with finesse. Can you do it with more than half your fuel still remaining? Can you do it without straying at all from the target perimeter? Can you get yourself into a stable elliptical orbit with the hole at one focus, so that you don't have to expend any more fuel at all? 16K, ROM Applesoft. Cassette, $14.95; Disk, $18.95.

□ **Apple-oids.** By Tom Luhrs, California Pacific (Davis, CA). Same old popular arcade game, except that those rocks coming at you have a curious fruitlike shape, with a chip out of the upper right shoulder. Big apples burst into medium-sized ones when you hit them, and the medium-sizers break up into little applettes that float around your screen, trying to blast you into oblivion.

Saucers enter the fray in two sizes: a big one that fires wildly in all directions and a small one that aims right for you. The big ship is a pretty easy target, notwithstanding his erratic course across the screen, but that little guy will nail you almost before you see him.

Every time you clear the screen the Apple-oids come back at you in larger numbers (up to a fourth-round maximum). You start with three ships at your disposal, and if you can accumulate ten thousand points before these three are destroyed, another ship joins your fleet.

Control of the game is via keyboard and paddle one. The paddle governs the rotation and thrust of your ships, while the Apple's number keys pull the trigger. Hitting a nonnumeric key jumps you into hyperspace—at your own risk. You may blow up on reentry, or you may relocate in a worse space than that from whence you fled.

The author has also designed a piece of hardware to plug into an Apple game port, which mimics the control panel of the actual arcade set up. Five buttons control left and right rotation, thrust, firing, and the leap into hyperspace.

On the flip side, as it were, is an earlier game by Luhrs, called *Chipout.* Like *Apple-oids,* it's a well-implemented remake of a familiar game—in this case of the old breakout genre. *Chipout* adds some spiffy touches: when you penetrate to the fifth row of bricks, the ball speeds up considerably; if you make it to the back wall, your bat shrinks to half its original size. And if you can clear the field with the five balls at your disposal, the game supplies a whole new field of bricks and lets you keep adding to your score. 32K, disk. $29.95.

Major League Baseball. By Stan Erwin, Color Software (Indianapolis, IN). This is the first sports program written for the Apple that actually attempts to replicate the performances of real professional athletes over the course of a full season.

Board games by APBA and Strato-Matic have been around for years and are the objects of reverence by their respective followers, so it's really a surprise that no one else has jumped on the bandwagon. *Major League Baseball* is a good start toward an excellent simulation of real-life major league baseball.

It takes into account several statistics for hitting and pitching and assigns an arbitrary defensive rating to each player based on his fielding performance of the prior year.

The player whose team is in the field can choose whether to pitch to a hitter or intentionally walk him. The player controlling the batting team has several more options, including sacrifices, hit-and-run plays, and steals when men are on base.

Most usual situations to be found on the diamond seem to be covered by the game; but one of the delights of baseball is that the unusual—triple plays, ejections for arguing with the umpire, and so forth—are commonplace enough to make the game interesting.

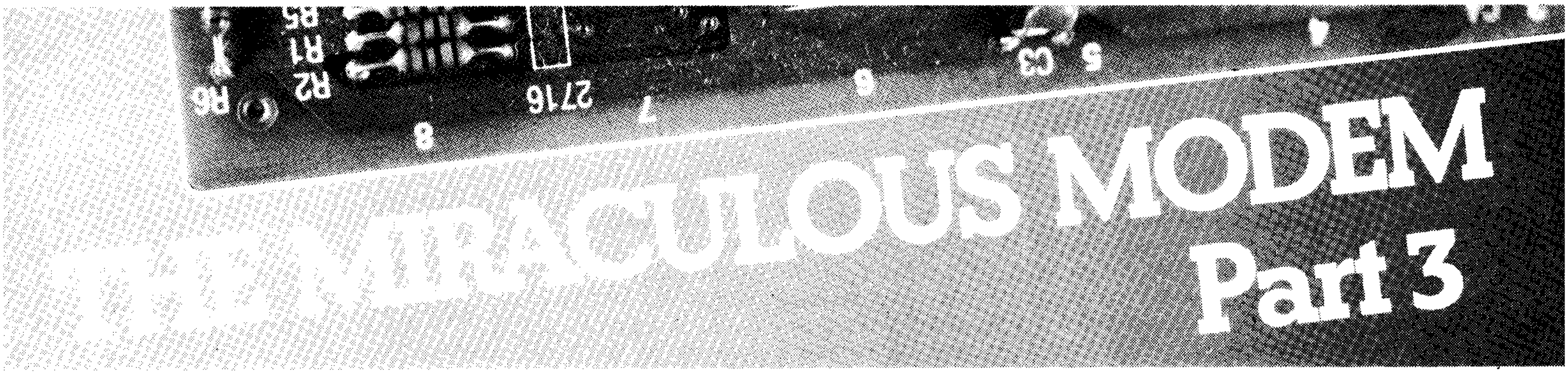
The board games have made provisions for these less common possibilities, but there seems no such provision in *Major League Baseball.*

The program falls down in two areas, one essential to the sports buff and one essential to the computer buff. For the sports buff, outs are merely announced without accounting for the kind of out made or the possibility for advancement by baserunners.

The computer buff will dislike the lack of animated graphics. A baseball diamond is displayed at all times, reflecting the current status of base runners. But there are no moving fielders and the baserunners appear mystically at their destination without running the bases.

The program provides two innovative utilities—one updates the statistics for each team at the end of each game and one permits you to update your teams with the following year's performances or to reflect trades.

The realistic performance of the players makes this an attractive package for baseball fans. Hopefully, Erwin will build on this solid base, making more use of the Apple to provide for additional realism. 48K, Applesoft ROM, disk. $25.



THE MIRACULOUS MODEM Part 3

BY JEFFREY MAZUR

Information Utilities

Just as we now rely upon our gas, electric, and telephone companies to supply their respective services, soon we will all become tied to another: the *information utility*. So much is being stored in computers today that it seems only natural to make some of this information available to the public— for a reasonable fee. All that is required for tapping this mountain of data is a terminal or microcomputer, a modem, and an account on one of the time-sharing utility systems. The two major information utilities available at present are the Source and CompuServe (also referred to as MicroNet).

The Source. Through the network of Prime computers located in Maryland, Source users can access the latest United Press International (UPI) newswires, the *New York Times* consumer data base, domestic and international airline schedules, restaurant guides, and much, much more. Also provided on the Source is a powerful electronic mail service through which you can send letters to other Source users. However, the most exciting feature of the Source is its interactive services that allow you to make real transactions such as buying merchandise or making travel arrangements.

The Source Telecomputing Corporation (formerly the Telecomputing Corporation of America) had a somewhat rocky start. Early users found many of the advertised services severely lacking or nonexistent. The system was prone to crashing, possibly ruining hours of work or even destroying important files stored on users' disks. As more and more people signed up on the network, response times from the system grew unreasonably long. For several months, there were rampant rumors that the Source was going bankrupt. But *Reader's Digest* saw the potential of this medium and purchased the ailing Source. With its financial problems solved, the Source has been steadily growing and improving. Just recently, the company has added a completely new and separate system to split up the load. New services and databases are constantly being added on both systems.

Joining the Source requires payment of a rather large ($100) initial hookup charge. This gets you a sign-on number and password to access the system. The actual data link is established via one of two independent telecommunications networks, Telenet or Tymnet, making the connection a local call for most users. Connect time on the Source is then charged at either the prime rate, 7 a.m. to 6 p.m. Monday through Friday, of $15 per hour, or the off-hour rate of $4.25 per hour. Very off hours—after midnight, local time—go for $2.75 per hour. Other services, including permanent disk storage, are extra. All charges incurred with your use of the Source are billed to a credit card account (VISA, Master Card, or American Express) that you must provide when you sign up. There is also a minimum charge of $10 per month.

After your application is processed, you'll receive the Source User's Guide, an identification number with a secret password, and a list of telephone numbers through which you can reach the Source. Getting connected to the Source is relatively simple once you've done it a couple times. Let's say that there's a Telenet access number in your local dialing zone. After calling the number, you receive a tone from the remote modem that your modem recognizes and establishes contact with. At this point, you're connected to Telenet and must tell it what computer you wish to talk to. In response to the "@" prompt, you type, "C 301 24" to contact the Source's System 10. If all goes well, you receive a greeting message from the Source along with a request to sign on. You now enter the letters ID followed by the sign-on code and password issued you by the Source. When the system is convinced that you're legit, it will inform you of the last time you signed on and then announce: "Welcome to the Source!"

With those words, the entire resources of this massive information utility are set before you. If you wish to check the latest UPI releases, simply type DATA DANEWS. This will bring in a description of how to find any items of interest. You can first choose whether you want international, national, state news. Then you further select from general, business, sports, and other categories. If there is a particular story you're looking for, it may be possible to find it directly be entering a few keywords, such as *space shuttle*. The information contained in this service is as up-to-the-date as you can get; it's available to you hours before it could appear in any newspaper.

After browsing through the UPI stories, you might decide to check the airline schedules to plan an upcoming trip you must make. Typing AIRSCHED-D (D for domestic) takes you into this part of the Source. By giving the departure and destination cities, you can receive a listing of all flights between the two. Having picked a flight, you can transfer to the travel agency service on the Source to purchase the tickets directly and have them sent to you. While you're at it, you can take care of the hotel reservations. And, if you happen to be going to New York or Washington, D.C., there's even a restaurant guide to these cities to find a nice place for dinner.

If you need a new piece of luggage for this trip, it might be worthwhile to check the discount shopping service for any special deals on such an item. Before signing off, you should see if anyone has sent you some mail (MAILCK), or perhaps you have a letter or two you would like to send to another Source user. Messages for the "establishment" can be left in the suggestion box.

CompuServe. The Ohio-based CompuServe makes their large DEC-10 mainframe computer available during off-peak hours, 6 p.m. to 5 a.m. local time plus all day Saturdays, Sundays, and holidays. The rate is $5 an hour, but the best part is that the initial sign-up fee is only $9. Access can be made directly from almost two hundred major cities or via the Tymnet network (an additional $2 per hour is then charged). There is no monthly charge if you do not use the system, so CompuServe will generally cost less than the Source. To top it off, they throw in 128K of disk storage on MicroNet at no extra charge. Like the Source, you must supply a credit card account for billing purposes.

Let's take a quick tour through CompuServe to get an idea of what it offers. After dialing the (hopefully) local access number and establishing connection, you type a control-C to get things started. This brings forth a request for your ID number and password. Having entered these correctly, you are logged on and informed of any important bulletins regarding use of the system. If there is any new electronic mail waiting to be read, you will be notified. Finally, you are presented with the main menu:

```
COMPUSERVE              PAGE CIS-1
COMPUSERVE INFORMATION SERVICE
1 NEWSPAPERS
2 FINANCE
3 ENTERTAINMENT
4 COMMUNICATIONS
5 COMPUSERVE USER INFORMATION
6 SPECIAL SERVICES
7 HOME INFORMATION
9 MICRONET PERSONAL COMPUTING
ENTER YOUR SELECTION NUMBER,
OR H FOR MORE INFORMATION.
```

The designation CIS-1, at the top of the screen, represents the page you're looking at. A page is roughly equal to one screenful of information. Most of CompuServe can be thought of as a library full of books. You can go directly to any page if you know its number or you can travel through the nested menus to get to the information desired. Using the menus is exactly like picking the right library to go to, looking in the card catalog for the right book, and checking the book's index to see what page you'll find the information you want on. Only it's a lot quicker on the computer!

One of the most useful services of CompuServe is its personal computing system, MicroNet. Although represented as just another choice on the main menu, MicroNet is really like a complete computer system within a computer system; and its features are tailored for the personal computing field. These features include several programming languages, many business and financial programs, and, of course, a wide variety of games. Some of the games even allow you to play against other MicroNet users in real time. Along this same line is one of MicroNet's most popular services: the CB Simulator. As the name implies, this allows you to talk to other users who are currently logged on. You select a "handle" by which the other users may recognize you and then pick one of forty channels to talk on. Whatever you type now will appear, preceded by your handle, on the screen of anyone listening on that channel. One channel may have as many as thirty people on at a time. Like its namesake, CB Simulator can get too crowded to be intelligible. Fortunately, at this time it seems that most people who come on the CB Simulator prefer just to listen in and not to talk.

If the only thing you like to play is the stock market, then the MicroQuote service is probably where you'll spend your time. Although there is a small extra charge for this service, you can gain instant access to current and past data on most stocks or bonds. Couple this with the financial advice offered in the CompuServe database, and you should improve your chances for picking a winner.

Another special feature on MicroNet is the Software Exchange. This is a collection of programs, many of which are written for the Apple II. These programs can be purchased and then downloaded into your computer over the phone. A special executive program is needed for this; this is also available from MicroNet, at no charge. In addition, this executive program provides screen formatting for forty columns, simple switching between the local (Apple operating system) and terminal modes, and also allows you to transfer files between your disk space on MicroNet and the Apple.

The Future. Both the Source and MicroNet are still in their infancy, yet they already provide an impressive amount of information and other services. As computers find their way into more and more homes, the need for these utilities will skyrocket. There will be a constant demand for increased services and improved reliability and efficiency. The private computer telecommunications network will rapidly expand to include banks, local merchants, medical help, other utilities, and much more.

Beginners' Corner

By Craig Stinson

This month, Softalk *begins a new column devoted specifically to the novice computer user. We assume here that the reader either owns an Apple, uses one on the job, or has been eyeing a friend's (and has borrowed said friend's copy of* Softalk*).*

Our intention is to provide the newcomer with a gentle introduction to the fascinations of personal computing. We will begin with a lesson in how to run some demonstration programs provided with the Apple, and, in subsequent columns, we will cover some of the elementary concepts of computing. One ongoing objective will be to define common computer terminology and jargon.

We believe that personal computing is no longer something only for the specialist or hobbyist. Personal computers will soon be as commonplace as radios and washing machines. We welcome that development, and we welcome to this corner of Softalk *the new Apple owner or Apple user.*

So let's fire up the Apple and see what it will do. If your system is not yet set up and ready to go, you'll need to consult chapter one of either the *Applesoft Tutorial* or the *Basic Tutorial,* depending on which of those two manuals came with your machine. When you're set, come back here.

At this point, you should have (1) a computer, (2) a TV set or monitor, (3) a disk drive, and (4) (we hope) a pair of game paddles. Apple used to supply game paddles with every computer sold, but they stopped doing this early in 1981. If your machine is brand new, you may be paddleless for the moment, in which case you won't be able to play one of the games that came with your system. But there are some other good things you can do and, when you want paddles, your local dealer should be able to sell you a pair; they're handy for a number of purposes—including games.

We realize there are some systems out there that don't have disk drives, although, if yours is relatively new or if you use it in a business setting, the overwhelming probability is that it does use a disk. For the purpose of this article, we'll presume that this is the case.

Turning On with a Beep. If everything is set up, the first thing you need to do is turn on your TV set or monitor so that

Glossary

Apple II—An Apple whose native language is Integer Basic. With additional hardware, it may also be equipped to understand Applesoft Basic.

Apple II Plus—An Apple whose primary built-in language is Applesoft Basic. With hardware modifications it can become fluent in Integer Basic as well.

Applesoft Basic—One of the two dialects of Basic available on the Apple. Also known as floating-point Basic, it has a larger vocabulary than Integer Basic and will do a number of things that Integer will not. Most commercial programs written in Basic for the Apple now are written in Applesoft.

Basic—An acronym for Beginner's All-purpose Symbolic Instruction Code, Basic is a friendly, general-purpose computing language developed at Dartmouth College in the 1960s. More to come on the topic of computer languages.

Control-C—Achieved by holding down the Control key while pressing C, control-C will halt the execution of most programs written in Basic.

Cursor—A prompt by which the Apple tells the user that it's expecting to receive communication from the keyboard. The cursor also shows the user where keyboard input is going to be displayed on the screen. The Apple's normal cursor is a blinking white square.

Integer Basic—The other Basic dialect available on the Apple. Integer is the older of the two Basics; the early Apples were all Integer machines, and hence some of the older software on the market requires Integer. As its name implies, it cannot compute with fractional numbers. Within the scope of its capability, however, it executes somewhat faster than Applesoft.

Prompt—Any symbol or message by which the computer tells the user what it's expecting next. For example, the closing bracket (]) and the right-pointing wedge (>) inform the user that the Apple is currently understanding Applesoft Basic or Integer Basic, respectively.

Return key—Located in the upper right corner of the keyboard, the return key is most commonly employed to tell the Apple that the user has finished inputting a line at the keyboard.

you'll have a picture when your computer comes on. Next, reach around to the back of the Apple with your left hand; just to the left of the power cord you will discover the first important landmark on your Apple: the on-off switch. As you will learn in due time, there's a reason why the switch was put back there in that clumsy, out-of-the-way part of the machine.

When you flip the switch on, various things will happen. The Apple's built-in speaker will beep once to let you know the system is running, and the power light in the lower left corner of the keyboard will illuminate. In the center of your screen you should see the Apple II logo. That power light, by the way, is only an indicator and not part of the keyboard per se. You can tap on it all day without affecting the status of your machine.

At this point, the little red "in use" light on your disk drive should also be lit.* You probably noticed some clacking noises coming out of the drive at first, followed by a steady whirring sound. The drive is spinning around, looking for something to read, so grab the disk marked "System Master" that came with your computer.

Hold the disk with your thumb over the label and your index finger directly on the other side of the disk. You'll find three places where the disk itself is visible through openings in its black cardboard case. The big hole in the center, called the hub, is where the spinning mechanism of the drive grabs the disk. The little hole to the side of the hub is a timing device to help the drive orient itself, and it's through that other, oblong-shaped opening that the drive actually reads information stored on the disk.

No Way To Treat a Floppy. It's important not to let your fingers stray onto any of these exposed areas of the disk. Some other things to avoid are leaving your disk on your dashboard in the noonday sun, setting it on your TV or monitor (magnetic fields can erase a disk), and bending it. It's a very good idea to keep disks in their protective envelopes when you're not using them.

If it's not open already, open the little door to your drive now, slide the disk in carefully, and close the door. Various things will happen now, depending on what species of Apple you have.

Delicious or McIntosh? As you may know, the Apple II comes in two main varieties, one called simply Apple II and the other called Apple II Plus. While both have the built-in intelligence to understand a computer language called Basic, they understand different dialects of it. The Apple II's native tongue is called Integer Basic (well, its native tongue is really binary, but we'll get into that later), while the Apple II Plus understands a somewhat more versatile dialect called Applesoft Basic.

It is possible for an Apple II to be taught Applesoft Basic by means of a piece of hardware (a circuit board or "card") inserted into the machine. And an Applesoft machine can similarly be made to understand Integer.

In subsequent columns, we'll have more to say about the differences between Apple IIs and Apple II Pluses and about computer languages in general, but for now it's important to know that if your computer is expecting to be addressed in Integer Basic it will display a symbol like this, >, next to a blinking white square. If it wants to be spoken to in Applesoft, it will display a closing bracket symbol,], next to that blinking white square.

The blinking square, by the way, is called a cursor, which is the Latin word for "runner." When you type on the Apple keyboard, the cursor runs around on the screen to show you where the next character is going to appear. The other symbol, the bracket or the wedge, is called a prompt, because it lets you know what language the Apple is set up to receive.

Spricht es Integer? If your system is supposed to be bilin-

*If the red light does not come on, and you see an asterisk on your screen, type 6, then hold down the control key (CTRL) with one finger while pressing P with another, then hit the key marked return. That should wake up your disk drive.

gual, you can test it by doing the following: If the computer is showing you an Applesoft prompt (the bracket), type INT and hit the key on the right side of the keyboard marked return. If your machine is wired to speak Integer as well as Applesoft, the Integer prompt will appear. If not, you'll get a beep and a message saying, "Language not available." Similarly, if your computer's native tongue appears to be Integer and you'd like to know whether it also speaks Applesoft, type FP and hit return. You'll get either the Applesoft prompt or an error message.

Getting error messages, by the way, is like falling down in the snow when you're learning to ski. It happens in the best of families, and you're never going to get anywhere without spending a certain amount of time in that condition. It's good to experiment, and nothing you can do at the keyboard, short of physical abuse (pouring coffee between the keys, for instance), will ever harm your computer. As you'll learn later on, there are some things you can do to damage data you may be working with, but the Apple itself is (on the outside) most impervious to any fooling around you may care to try.

Opening the Cookie Jar. Now that you know what languages you have available, you're ready to run some programs. To see what's on the System Master disk, type CATALOG and hit return. *Voila,* a list of the first eighteen programs on the disk. Hit any key at all and the list will scroll upward to reveal the remainder of the disk's contents. At the end of the list, you'll find your familiar language prompt.

Each item in the list is accompanied by an asterisk, a code letter, and a three-digit number. The asterisk signifies that the file is "locked," which means that if for some reason you wanted to wipe one of these programs off the disk, you would have to go to a little extra trouble to do it.

The letter code identifies the file type. Programs with an A beside them are written in Applesoft and can only be run if you have Applesoft available on your machine. Similarly, an I program can only be run if you have Integer available.

If your machine speaks both dialects of Basic, you can run either I or A programs. You do not have to have the appropriate prompt showing. If your machine is showing the Applesoft prompt and you want to run an Integer program, go ahead and run it and the computer will switch languages automatically.

Programs marked with a B are written in machine language (the B stands for binary), about which we'll have more to say when we get to a general discussion of programming languages. The important thing for now is that both Apple IIs and Apple II Pluses can run B programs.

Finally, the three-digit number between the letter code and the program name is an indicator of the size of each program.

Your First Fix. If you have Applesoft Basic available on your machine and you have a set of game paddles hooked up, you're ready to play *Little Brick Out.* Type RUN LITTLE BRICK OUT and hit return, and the computer will provide you with instructions and prompts to get you going. *Little Brick Out* is addicting, and, chances are, once you start playing, you won't want to read any more of this, so we'll see you next month!

If yours is an Integer-only machine, type RUN APPLEVISION and enjoy a demonstration of the Apple's sound and high-resolution graphics capabilities. This program, by the way, will repeat endlessly until you tell it to stop (or turn your machine off). So when you've had enough, hold down the control key (marked CTRL, on the left, right above shift) with one finger while you press C with another. This operation, commonly known as "hitting control-C," will interrupt most programs that run in Basic. *Applevision* will stop when you hit control-C, but you won't see your cursor again and, depending on when you interrupt the program, you may not even see the Integer prompt. That's because the Apple is still in its high-resolution graphics mode. To return it to its normal state so you can see what you type next, enter the letters TEXT and hit return. You may not see those letters until you hit return, but don't let that stop you.

Up a Computer without a Paddle? If your computer knows Applesoft, but you don't have a set of game paddles, try running *Brian's Theme,* a graphics demo program that will paint intriguing moire patterns all over your TV or monitor screen. These patterns are produced with the help of Apple's random number generator, which means that no two executions of the program will be quite the same.

This program, like *Applevision,* will run forever unless you stop it. Unlike *Applevision,* however, which simply repeats the same material ad infinitum, this one constantly changes. If you feel like pondering that old philosophical question about whether a tree falling in the forest makes any noise if no one's there to hear it, shut off your monitor, leave the computer on, and go to bed. Those little electrons will keep running around in their little circuits, the Apple's random number generator will keep doing its thing, and when you turn the monitor back on in the morning you'll find your computer is still spinning out pretty patterns for you to enjoy.

You might be interested to know that your Apple uses about as much electricity as a forty-watt light bulb. So if there ever is any reason to let a program run all night, it won't do much to your electric bill. Your TV set takes a lot more power.

Make Sure You're Not Off-Color. In addition to these programs, the System Master disk also has programs in each of the Basic dialects specifically for tuning in your color TV or color monitor to the Apple's color graphics. These are *Color Demo* for Integer and *Color Demosoft* for Applesoft. There's also a nifty Integer game called *Animals* and a useful *Phone List* in Applesoft that will allow you to store up to a hundred names and phone numbers. These last two programs, however, both require that you have available a blank, initialized disk, and we'll get into what all that means next time.

What about all the binary programs on the System Master? These, and the other Basic programs we've not mentioned yet, are called utility programs; their utility will become more apparent before too long. We'll talk more about some of them next month. See you then.

THE PASCAL PATH

By Jim Merritt

The Path Becomes a Railroad

The conscientious use of proper spelling and grammar is important in English, essential in Pascal, and difficult in either, at least for the beginner. Although in English you can break a lot of rules and still communicate your ideas with reasonable effectiveness, every Pascal computer program must be *completely correct* and free of syntax errors, or it will not even compile, much less execute.

Special syntax charts ease the burden of preparing syntactically correct programs and have been employed by Pascal programmers ever since the language was introduced. These charts have been nicknamed *railroad diagrams,* because they often resemble maps of the track layout at a railroad yard. When taken together, they amount to a complete definition of Pascal and, in a very real sense, reveal a great deal about how the compiler works in recognizing and translating the various parts of your programs.

We'll spend this installment discussing syntax charts, because, once you know the simple rules for reading them, you'll be able to use them to answer all your own questions concerning how to form correct Pascal statements and programs. This will free this column to concentrate on showing you methods of using the language to express your ideas. We won't have to waste time and space with long, detailed presentations of grammar.

Reading a syntax chart involves little more than putting yourself in the place of the compiler. When you give it a source file name, it expects to find text in the file that corresponds to a *compilation.* Figure 1 is a syntax chart for a compilation. The compiler's scan of the source is analogous to tracing the syntax diagram from its beginning in the upper lefthand corner to its end in the upper righthand one. The trace proceeds along the *scan-line* in the direction of the arrowheads.

When the scan-line is interrupted by a box, an oval, or a circle, this indicates that the compiler expects to recognize a certain Pascal object at that point in the source text. Objects in ovals or circles correspond to actual words and characters that the compiler should find in the text. For example, in Figure 1, the period and semi-colon are encircled. This means that the compiler must encounter a period or semi-colon in the text at the respective point; otherwise the compilation is not valid. An object in a rectangle is defined by another chart, and you should temporarily skip to that chart, whose title is the word or phrase enclosed by the rectangle, in order to verify the object. You will eventually see charts named *program* and *unit,* although you don't have to worry about them right now. (You already have an idea about what a program is, and we'll be spending most of our time expanding that concept. Units are similar to programs, but won't be examined in the *Pascal Path* for quite some time.)

Often, a chart's scan-line will fork, indicating that Pascal permits exactly one of two or more kinds of objects to exist at a

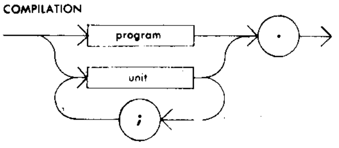


Figure 1.

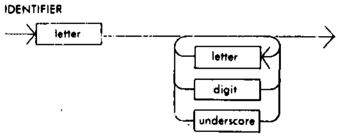


Figure 2.

certain point in the text. Figure 1 indicates, for example, that either exactly one program *or* one or more units, *but not both,* may be included in a compilation. Notice that, if a semicolon is placed after a unit (whatever that may be), another unit *must* follow. Finally, all compilations *must* conclude with a period.

You can use the compilation chart to practice as a grammarian, without knowing the details of what constitutes a program or a unit. For purposes of discussion and typography, let's represent the program object as [PROGRAM] and the unit object as [UNIT]. Here are three legal compilations, according to Figure 1:

[PROGRAM].
[UNIT].
[UNIT]; [UNIT]; [UNIT].

Here are some incorrect compilations:

[PROGRAM]
[PROGRAM]; [PROGRAM].
[UNIT][UNIT];
[UNIT]; [PROGRAM];
[UNIT];.

Don't just take my word for it. Use the chart to convince yourself that errors do in fact exist in the latter examples. If what you see in an example doesn't correspond to what you expect to see by tracing the syntax chart, the example is wrong. The same may be said of any code that *you* write.

Figure 2 presents a chart for the object *identifier.* The box *digit* corresponds to exactly one of the digits 0 through 9; the box *letter* corresponds to exactly one of the upper or lower case alphabet characters. After passing through exactly one of these, you go on to the end of the diagram. Space limitations prevent our showing the latter three charts. You may wish to trace these diagrams on your own. The only reason the *underscore* has its own chart is that it might be mistaken for a hyphen or dash.

Play the role of the compiler, using the syntax chart in figure 2 and those you may have traced to attempt to recognize these example identifiers, some of which are legal, some illegal:

Azimuth
FindCustomer
7Up
GetNumber
Jack&Jill
String2Integer
CubeRoot
GeneralLedger
Accts_Payable
Post Accts
Program5
VerifyKey

Which of the above fail to qualify as identifiers? You probably don't need the charts to tell, but use them anyway, for practice. In the unlikely event that you get stuck for an answer, refer to the section on identifiers in the second installment of this column (March 1981). Nearly all these examples were noted as legal or illegal there.

Appendix F in the Apple Pascal *Language Reference Manual* includes a complete set of railroad diagrams for Apple Pascal. From now on, whenever I introduce a new kind of Pascal construction, I will also show its syntax chart. Now and then, Apple's charts will differ slightly from mine, as was the case this time with the one for compilation. In such instances, assume that mine is correct, even though Apple's is probably just as valid. Careful comparison will show that, except for typographical errors in either set, Apple's charts depict exactly the same syntax as my own.

Your standing assignment, whenever a new railroad diagram is given, is to trace it out as completely as you can, using any examples I supply, as well as your own. I guarantee that your progress along the *Pascal Path* will be much quicker and easier if you make a habit of "taking the train."

Apple Pascal Special Keystrokes, Group 1

During several months, the Pascal Path discussed the nature and use of Apple Pascal prompt lines. This chart has had to wait its turn for space; now you can reunite it with the prompt lines themselves, which appeared in May, for a fine reference.—MCT

Prompt Line → Key or ↓ Couplet	System Main Prompt Line	Editor Prompt Line	Editor Insert Prompt Line	Editor Delete Prompt Line	Editor Exchange Prompt Line
C	Invokes Pascal compiler.	Invokes copy mode.*	Taken as literal text.	Ignored.	Taken as literal text.
D	*Most often:* Invokes Pascal compiler*.	Invokes delete mode.	Taken as literal text.	Ignored.	Taken as literal text.
E	Invokes Editor.	Ignored.	Taken as literal text.	Ignored.	Taken as literal text.
I	Reinitializes Apple Pascal System (similar to bootstrap process). *Be careful!* *	Invokes insert mode.	Taken as literal text.	Ignored.	Taken as literal text.
X	User-specified program is executed.	Invokes exchange mode.	Taken as literal text.	Ignored.	Taken as literal text.
<ESC>	Ignored.	Ignored.	*Panic button.* Insertion is aborted. All material inserted during current insert session is thrown away. Screen is restored to its previous condition.	*Panic button.* Deletion is aborted. Any material deleted during current insert session is restored, along with the previous screen condition.	*Panic button.* Aborts exchange. Original text (prior to exchange) is restored, along with previous screen condition.
<CTRL—C>	Reinitializes system. *Be careful!* *	Ignored.	Completes insertion. Inserted text becomes permanent part of main buffer and screen display.	Completes deletion. Deleted text is permanently removed from main buffer and screen display.[2]	Completes exchange. Text replacement is permanent.
<—	Ignored.	Moves cursor one character position to the left (does *not* erase). (At beginning of line, cursor jumps up to last position in previous line.)	Moves cursor one character position to the left, erasing the character in that position. (Can erase only material entered during current insert session.)	Moves cursor one character position to the left. If moving away from delete session starting position, erases character *. If moving toward delete session starting position, restores an erased character *.	Moves cursor one character position to the left, restoring the character that occupied the position before current exchange session.
—>	Ignored.	Moves cursor one character position to the right. (At end of line, cursor drops to first position on next line.)	Taken as literal text, but the question mark (?) is displayed since this character is normally invisible.	Moves cursor one character position to the right. If moving away from delete session starting position, erases character.* If moving toward delete session starting position, restores an erased character.*	Taken as literal text, but the question mark (?) is displayed since this character is normally invisible.
<CTRL—O>	Ignored.	Cursor moves up to same column position in previous display line (if any).	Taken as literal text, but the question mark (?) is displayed since this character is normally invisible.	Cursor moves up to same column position in previous display line (if any). Text between old and new cursor positions is erased.	Taken as literal text, but the question mark (?) is displayed since this character is normally invisible.
<CTRL—L>	Ignored.	Cursor moves down to same column in next display line (if any).	Taken as literal text, but the question mark (?) is displayed since this character is normally invisible.	Cursor moves down to same column in next display line (if any). Material between old and new cursor positions is erased.	Taken as literal text, but the question mark (?) is displayed since this character is normally invisible.
<CTRL—X>	Ignored.	Ignored.	Erases current input line, positioning cursor at end of previous line. Can erase all but first line entered during current insert session.	Ignored.	Taken as literal text, but the question mark (?) is displayed since this character is normally invisible.
<RETURN>	Ignored.	Moves cursor to beginning of next text line (if any).	Completes text line. Cursor moves to beginning of next display line (plus remembered indentation, if any), and insertion continues there.	Cursor moves to starting position in next display line. All text between old and new cursor positions is erased.	Ignored.

All prompt lines are displayed on the top line of the screen.
* A key sequence or result that has not been discussed.

[1] All keys are understood by Apple Pascal in upper or lower case.
[2] Display collapses together to fill any holes caused by deletion.

MARKETALK

News

from page 47

save/create files, and printout modes. System is update of earlier *Statistics* package. 48K, DOS 3.3. $29.95.

□ A utility for their own *Fast Floating Point* board comes from **Computer Station** (Saint Louis, MO). The patch, dubbed *Applefast*, is loaded into your RAM card upon booting and allows fast calculations of transcendental functions utilizing the AM9511 arithmetic processor on the *FFP* board. The 3.3 utility is $25 for registered owners of the *FFP* board.

□ Communicate with other Apples and Western Union with two packages from **Microcom** (Boston, MA). *Micro Courier* allows you to send correspondence, graphs, charts, programs, or *VisiCalc* reports over phone lines to an awaiting Apple on the other end. Transmission is automatic and twenty-four hours. *Micro Telegram* hooks Apple up with Western Union, letting you send and receive Telex, TWX, international cables, and mailgrams, and giving access to Western Union's data base and news and information source, Infomaster. Both require two disk drives, D.C. Hayes Micromodem, 48K, DOS 3.3, Basic. $250 each.

□ The *AI13 Analog Input System* from **Interactive Structures** (Bala Cynwyd, PA) makes random access readings in monitoring and measurement applications. The analog to digital conversion system can set, by software command, its full-scale range to any of eight different ranges. Monitor program, simulated chart recorder and three other sample programs included on accompanying disk. $425. Interactive Structure's *PR12* printing system interfaces with Apple and Epson *MX80* mechanism and features hi-res graphics capabilities. *PR12* offers 6 character sizes from 5 to 16½ per inch, software-definable characters and symbols, and tabs for column printing and indenting. $630.

□ **Science Research Associates'** (Chicago, IL) *Computer Discovery Program* teaches computer literacy to junior high and high school students. Package of two disks and twenty workbooks takes pupil through a history of computing, essentials of programming while getting to know an onscreen robot, ins and outs of software and hardware, and a study of the social effects of computerization. 48K, DOS 3.3, Applesoft. $184.25 at either level.

□ Bus and music systems are available from **Passport Designs** (La Honda, CA). *Softbus* is slot independent and routes bus line assignments via three three-foot ribbons (control, data, address) to your breadboard. Modifications of any circuit designs can be done without getting into or turning off Apple. $49.95. *Soundchaser* music system provides three and six voice capability with a polyphonic keyboard and interface card ($650), a three voice synthesizer card ($350), and accompanying software. Keyboard has four octaves; each synthesizer has three independent voice modules, each of which consists of a square/sawtooth waveshaper, a dynamic loudness amplifier, a twenty-four dB/octave low pass resonant filter, and an audio oscillator. Software features sequencer, edit, and system master modes; sequencer can store from two to eight minutes of note, chord, and rhythm information. 48K, DOS 3.2 or 3.3. Three voice system, $1,000; six voice system, $1,350.

□ The *Apple-Cat II* large scale integrated modem from **Novation** (Tarzana, CA) provides Baudot and ASCII code in a single add-on device, permitting contact with ASCII-based machines and Baudot machines used by the deaf. Modem can operate at speeds from fifty to twelve hundred baud; auto dials, redials, answers and disconnects phone calls; and decodes standard tones for remote control of Apple from a Touch Tone phone. Optional program chip makes modem compatible with Basic, Pascal, and CP/M. $389.

TRADETALK

from page 60

that fast-growing distributor of software has added nine more companies to their line and, in the process, has expanded into hardware. New accounts are **Software Publishing Corp., DataSoft, Denver Software, Sentient Software, BudgeCo, Sir-Tech, Highlands Computer Services, Microsoft,** and **TG Products.** Last month, eight of *Softalk*'s Top Ten and seventeen of the Top Thirty Bestsellers were products distributed by Softsel.

□ With the creation of the position of dealer sales coordinator, **Automated Simulations** (Mountain View, CA) found someone to fill it. She is **Kathy Carlson**, formerly the company's customer service representative. As she did in her former capacity, Carlson will provide support for retail outlets, and also inform dealers and distributors on new products and their availability. She will devise retail sales and promotional aids and service programs for distributors and dealers.

□ **Personal Software** (Sunnyvale, CA) went no farther than their public relations firm, **Regis McKenna**, to find their new manager of product marketing. **Richard Melmon** has been a Regis McKenna vice president for two years; now he will oversee Personal's strategic market planning, product marketing, and its advertising and public relations ventures. Melmon has a head for both the technical and business worlds, as evidenced by his bachelor's degree in physics from UC Berkeley and a master's degree in business administration from Stanford University.

□ In order to keep a pulse on software innovations, **Microsoft Consumer Products** (Bellevue, WA) has found someone to help in this endeavor. **Alan M. Boyd**, as products development manager, will interface with independent software developers to make and license new packages to be marketed by the company. A native of Scotland, Boyd has been **Hayden Book Company**'s business software editor, **Muse Software**'s project manager, and a senior audio engineer with **Maryland Sound Industries**.

□ **MicroPro International** (San Rafael, CA) and **Microsoft** have come to an agreement: MicroPro will release a version of their *WordStar* word processor that will function solely with Microsoft's *SoftCard*. In effect, this will introduce Apple owners to one of the most highly regarded word processors in the industry. This joint marketing agreement sets Microsoft as distributor of the *WordStar/SoftCard* package.

□ A pact of another sort has been reached by **Micro Focus** (Santa Clara, CA) and **Apple Computer Inc.** Apple's dealer network is now carrying Micro Focus's *CIS COBOL* package, which consists of the ANSI '74 COBOL compiler for Apples with a CP/M environment, and a COBOL source-code generator. An initial shipment to the more than 1,000 Apple dealers has been made, with more shipments in the offing.

□ Second fiscal quarter 1981 sales of $78.8 million and a net income of $9.2 million, both 3.3 times the number attained a year ago, was recently announced by **Apple Computer Inc.** The number of common and common equivalent shares shot up in twelve months from 47.8 million to 57.3 million, with earnings per share now at $.16, as compared with $.06 this time last year. Growth from previous quarter was also in evidence: a 16 percent rise from first quarter's $67.6 million sales mark, a net income increase of 24 percent from $7.4 million, and earnings per share up 14 percent from the previous $.14. . . . Apple has set plans for addition of 170,000 square feet to the Dallas, Texas, manufacturing facility, and the July opening of a leased Singapore manufacturing plant that will be eventually replaced by a 100,000 square-foot plant now being built.

The Double Life of Lord British

Softalk photo

If you were to wander through the park in Austin, Texas, on a mild weekend afternoon, you might stumble across a scene from another age, a medieval battlefield peopled by full-plate-armored warriors and fencers. Among the fencers, you might see a young man of slight frame, delicately mustached, with large, gentle eyes and fine sandy hair.

Should you follow that youth, strike up a conversation, and ask his father's occupation—expecting king, lord, tinker, barkeep—you'd be told, anachronistically, "Astronaut."

The young man—eighteen to be exact and just finishing his sophomore year in college—is Lord British, creator of *Akalabeth* and, most recently, *Ultima. Ultima* combines the two lives of Lord British: it goes from the medieval times he loves and studies with the Society for Creative Anachronisms to the Space Age in which his father is a pioneer, and beyond.

Lord British, better known in the Barony of Bryn Gwald, Kingdom of Ansteorra, as Shamino, and in the world of space as Richard Garriott, has a mind equally at home in and equally delighted with one world as with the other. He is certainly the only microcomputer programmer to ply his trade while wearing headgear from the age of chivalry.

The mystery concerning Lord British's identity began when he started fooling around with Dungeons and Dragons-type games while working for a Texas Computerland. He signed these efforts Lord British and passed them out to friends for testing, until one caught the attention of Al Remmers, who offered to publish it.

Winners of the "Who Is Lord British?" contest were announced in *Softalk,* May 1981. No one gave a correct identification, which would have read something like this (in the order the clues were given):

"Lord British is a student at the University of Texas in the state of Texas (state of friendship). His home is in Houston (site of NASA's Space Flight Center). He works at the Computerland store on El Camino Real (the King's Highway) in the city of Clear Lake, Texas. And his father was an astronaut on Skylab."

His father is Owen Garriott, an astronaut aboard Skylab I who's scheduled to return to space with the first payload-bearing Skylab.

Softalk Presents The Bestsellers

It's a script only a B-movie producer could love.

The Kid had shown real creative talent. His early works were rated favorably by the critics and the masses alike. It looked like he would join the pantheon of greats at an early age.

Then something unexplained happens. Maybe he loses the creative touch. Maybe he is just distracted. He rejects his patron. Nobody hears or thinks much about him for a while.

Finally, he goes back to his art. It takes an agonizing nine months for his next work. He finishes and nervously awaits the popular reaction to his first creation done without a sponsor. Naturally, it's a big hit and The Kid lives happily ever after.

If you made the movie today, you might call it *Raster Blaster*, the name of The Kid's creation.

Seldom has a creative hiatus been so dramatically ended as Bill Budge ended his, with the finest display of state-of-the-art craftsmanship seen yet. From coast to coast, Apple retailers are asking in awe, "What in the world can he do for an encore?"

Business 10

This Month	Last Month	
1.	1.	*VisiCalc*, Software Arts/Dan Bricklin and Robert Frankston, Personal Software
2.	2.	*DB Master*, Alpine Software/Stanley Crane and Jerry Macon, and Barney Stone, Stoneware
3.	4.	*Apple Plot*, Apple Computer Inc.
4.	3.	*Apple Writer*, Apple Computer Inc.
5.	10.	*Supertext II*, Ed Zaron, MUSE
6.	—	*BPI General Ledger*, John Moss and Ken Debower, Apple Computer Inc.
7.	8.	*Personal Filing System*, John Page, Software Publishing Corporation
8.	5.	*Apple PIE*, Tom Crossley, Programma
9.	—	*SuperScribe*, David Kidwell and Jeff Schmoyer, On-Line Systems
10.	6.	*Data Factory*, Bill Passauer, Micro Lab
	7.	*Information Master*, James A. Cox and Stephen M. Williams, High Technology

Home/Hobby 10

This Month	Last Month	
1.	2.	*DOS 3.3*, Apple Computer Inc.
2.	1.	*Typing Tutor*, Image Producers, Microsoft
3.	3.	*DOS Tool Kit*, Apple Computer Inc.
4.	6.	*Data Capture*, David Hughes and George McClelland, Southeastern Software
5.	4.	*Home Money Minder*, Bob Schoenburg and Steve Pollack, Continental Software
6.	5.	*LISA Assembler*, Randy Hyde, Programma
7.	—	*ASCII Express II*, Bill Blue, Southwestern Data Systems
8.	—	*Financial Management System II*, Dennis Jarvis, D. R. Jarvis Computing
	8.	*Bill Budge's 3-D Graphics Package*, Bill Budge, California Pacific
	—	*DOS Plus*, Mike McLaren, Sensible Software

The Bestsellers

Budge and *Raster Blaster* set one standard that cannot be beaten, only equaled. *Raster Blaster* is the first program ever to come from a new software publisher and make the Top Thirty in its first month of distribution—actually it was less than a month and it seems clear the program would have made the top spot had it been generally available April 1.

As it was, *Space Eggs* held onto the number one slot, barely beating out a *VisiCalc* revived by the introduction of a DOS 3.3 version. *Raster Blaster* had to settle for third.

For the second straight month, the Top Thirty list proved more volatile than it had been earlier. The change this time—in a month generally marked by an upturn in business at the retail level—was marked by a 50 percent increase in nonentertainment programs on the list and the improved standing of some of the business software already on the list.

In addition to *VisiCalc* closing in on *Space Eggs*, *DB Master* leaped to fifth, *DOS 3.3* edged into tenth, and *DOS Tool Kit* rose to fifteenth. New entrants in the Top Thirty included *Data Capture, Personal Filing System, Supertext II*, and *BPI General Ledger*, now distributed by Apple.

Seemingly poised to join that group is *VisiPlot*, which made a strong run for the Business Ten, although not in widespread distribution for most of April.

In the entertainment area, Broderbund maintained a strong competitive position with *Snoggle* moving from ninth to fourth, *Alien Rain* sitting in sixth, and *Alien Typhoon* nudging into the list in a tie for twenty-fourth.

Ken and Roberta Williams made April a clean sweep—all three of their hi-res adventures made the Top Thirty.

Other than *Raster Blaster*, the only newly released program that made the Top Thirty was *Olympic Decathon* from Microsoft.

Apple-franchised retail stores representing approximately 8 percent of all sales of Apple and Apple-related products volunteered to participate in the poll.

Respondents were contacted early in May to ascertain their sales leaders for the month of April.

The only criterion for inclusion on the list was number of sales made—such other criteria as quality of product, profitability to the computer retailer, and personal preference of the individual respondents were not considered.

Respondents in May represented every geographical area of the continental United States as well as Hawaii.

Results of the responses were tabulated using a formula that resulted in the index number to the left of the program name in the Top Thirty listing. The index number is an arbitrary measure of relative strength of the programs listed. Index numbers are correlative only for the month in which they are printed; readers cannot assume that an index rating of 50 in one month represents equivalent sales to an index number of 50 in another month.

Probability of statistical error is plus-or-minus 5.4 percent, which translates roughly into the theoretical possibility of a change of four points, plus or minus, in any index number.

The Top Thirty

This Month	Last Month	Index	
1.	1.	99.56	*Space Eggs*, Nasir, Sirius Software
2.	2.	95.45	*VisiCalc*, Software Arts/Dan Bricklin and Robert Frankston, Personal Software
3.	—	76.36	*Raster Blaster*, Bill Budge, BudgeCo
4.	9.	55.80	*Snoggle*, Jun Wada, Broderbund Software
5.	11.	55.22	*DB Master*, Alpine Software/Stanley Crane and Jerry Macon, and Barney Stone, Stoneware
6.	3.	51.69	*Alien Rain*, Tony Suzuki, Broderbund Software
7.	6.	49.05	*Flight Simulator*, Bruce Artwick, SubLogic
8.	10.	42.88	*Hi-Res Adventure #2: The Wizard and the Princess*, Roberta and Ken Williams, On-Line Systems
9.	7.	39.65	*Warp Factor*, Paul Murray and Bruce D. Clayton, Strategic Simulations
10.	13.	37.89	*DOS 3.3*, Apple Computer Inc.
11.	4.	31.72	*Zork*, Mark S. Blank, Timothy Anderson, Bruce Daniels, P.D. Leblins, Scott Cutler, and Joel Berez/Infocom, Personal Software
12.	5.	30.25	*Sargon II*, Dan and Kathe Spracklen, Hayden
13.	15.	29.66	*ABM*, Silas Warner, MUSE
14.	12.	29.37	*Typing Tutor*, Image Producers, Microsoft
15.	19.	27.02	*DOS Tool Kit*, Apple Computer Inc.
16.	—	26.43	*Hi-Res Adventure #0: Mission: Asteroid*, Roberta and Ken Williams, On-Line Systems
17.	—	23.79	*Olympic Decathlon*, Tim Smith, Microsoft
	21.	23.79	*Apple Plot*, Apple Computer Inc.
19.	18.	22.03	*Apple Writer*, Apple Computer Inc.
	—	22.03	*Data Capture*, David Hughes and George McClelland, Southeastern Software
21.	23.	19.09	*The Prisoner*, David Mullich, Edu-Ware Services
	—	19.09	*Supertext II*, Ed Zaron, MUSE
23.	25.	18.80	*Creature Venture*, Butch Greathouse and Garry Reinhardt, Highlands Computer Services
24.	30.	18.50	*Asteroid Field*, Jim Nitchals, Richard Moore, and Barry Printz, Cavalier Software
	—	18.50	*Alien Typhoon*, Tony Suzuki, Broderbund Software
26.	—	17.92	*Hi-Res Adventure #1: Mystery House*, Ken and Roberta Williams, On-Line Systems
	—	17.92	*BPI General Ledger*, John Moss and Ken Debower, Apple Computer Inc.
28.	—	17.33	*Personal Filing System*, John Page, Software Publishing Corporation
29.	8.	16.74	*Phantoms 5*, Nasir, Sirius Software
30.	22.	14.69	*Apple PIE*, Tom Crossley, Programma

We've Moved!

Softalk was splitting its seams, and so was its office. So we've got a new heavier cover to help keep the magazine together and a new bigger office to help keep our heads together. Our new address is:

SOFTALK
11021 Magnolia Boulevard
North Hollywood, CA 91601

Our telephone number for editorial, advertising, and business remains 213 980-5074. But circulation has a new telephone number:

213 980-5099

This number, 213 980-5099, is the number to call with any questions about your subscription to *Softalk*: changes of address, signups, where's your issue from last month?, and so on.

If you're a new Apple owner, or an old one just discovering *Softalk*, here's a chance to catch up on what you missed.

And, if you know some Apple owners who aren't receiving *Softalk*, you can get your back issues free. For each Apple owner not yet receiving *Softalk* whose name, address, and Apple serial number you send to *Softalk*, you'll receive a back issue of your choice at no charge. All issues are available except November and December.